

API Documentation

To create a client we advise you to read the [Moodle web service client documentation](#)

[Print all](#)

`auth_email_get_signup_settings`

Get the signup required settings and profile fields.

Arguments

Response

General structure

```
object {
    namefields list of (
        string //The order of the name fields
    )passwordpolicy string Optional //Password policy
    sitepolicy string Optional //Site policy
    sitepolicyhandler string Optional //Site policy handler
    defaultcity string Optional //Default city
    country string Optional //Default country
    profilefields Optional //Required profile fields
    list of (
        object {
            id int Optional //Profile field id
            shortname string Optional //Profile field shortname
            name string Optional //Profield field name
            datatype string Optional //Profield field datatype
            description string Optional //Profield field description
            descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
            categoryid int Optional //Profield field category id
            categoryname string Optional //Profield field category name
            sortorder int Optional //Profield field sort order
            required int Optional //Profield field required
            locked int Optional //Profield field Locked
            visible int Optional //Profield field visible
            forceunique int Optional //Profield field unique
            signup int Optional //Profield field in signup form
            defaultdata string Optional //Profield field default data
            defaultdataformat int //defaultdata format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
            param1 string Optional //Profield field settings
            param2 string Optional //Profield field settings
            param3 string Optional //Profield field settings
            param4 string Optional //Profield field settings
            param5 string Optional //Profield field settings
        }
    )recaptchapublickey string Optional //Recaptcha public key
    recaptchachallengehash string Optional //Recaptcha challenge hash
    recaptchachallengeimage string Optional //Recaptcha challenge noscript image
    recaptchachallengejs string Optional //Recaptcha challenge js url
    warnings Optional //List of warnings
    list of (
        /warning
    )object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```



XML-RPC (PHP structure)

```

Array
(
    [namefields] =>
        Array
            (
                [0] => string
            )
    [passwordpolicy] => string
    [sitepolicy] => string
    [sitepolicyhandler] => string
    [defaultcity] => string
    [country] => string
    [profilefields] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [shortname] => string
                            [name] => string
                            [datatype] => string
                            [description] => string
                            [descriptionformat] => int
                            [categoryid] => int
                            [categoryname] => string
                            [sortorder] => int
                            [required] => int
                            [locked] => int
                            [visible] => int
                            [forceunique] => int
                            [signup] => int
                            [defaultdata] => string
                            [defaultdataformat] => int
                            [param1] => string
                            [param2] => string
                            [param3] => string
                            [param4] => string
                            [param5] => string
                        )
            )
    [recaptchapublickey] => string
    [recaptchachallengehash] => string
    [recaptchachallengeimage] => string
    [recaptchachallengejs] => string
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="namefields">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="passwordpolicy">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="sitepolicy">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="sitepolicyhandler">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="defaultcity">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="country">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profilefields">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="shortname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="datatype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="categoryname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="required">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="locked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="forceunique">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="signup">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="defaulldata">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="defaultdataformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="param1">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param2">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param3">
```



```

        <VALUE>string</VALUE>
    </KEY>
    <KEY name="param4">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="param5">
        <VALUE>string</VALUE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="recaptchapublickey">
    <VALUE>string</VALUE>
</KEY>
<KEY name="recaptchachallengehash">
    <VALUE>string</VALUE>
</KEY>
<KEY name="recaptchachallengeimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="recaptchachallengejs">
    <VALUE>string</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



Restricted to logged-in users

No

Callable from AJAX

Yes

auth_email_signup_user

Adds a new user (pending to be confirmed) in the site.

Arguments

username (Required)

Username

General structure

```
string //Username
```

XML-RPC (PHP structure)

```
[username] => string
```

REST (POST parameters)

```
username= string
```

password (Required)

Plain text password

General structure

```
string //Plain text password
```

XML-RPC (PHP structure)

```
[password] => string
```

REST (POST parameters)

```
password= string
```

firstname (Required)

The first name(s) of the user

General structure

```
string //The first name(s) of the user
```

XML-RPC (PHP structure)

```
[firstname] => string
```

REST (POST parameters)

```
firstname= string
```

**lastname (Required)**

The family name of the user

General structure

```
string //The family name of the user
```

XML-RPC (PHP structure)

```
[lastname] => string
```

REST (POST parameters)

```
lastname= string
```

email (Required)

A valid and unique email address

General structure

```
string //A valid and unique email address
```

XML-RPC (PHP structure)

```
[email] => string
```

REST (POST parameters)

```
email= string
```

city (Default to "")

Home city of the user

General structure

```
string Default to "" //Home city of the user
```

XML-RPC (PHP structure)

```
[city] => string
```

REST (POST parameters)

```
city= string
```

country (Default to "")

Home country code

General structure

```
string Default to "" //Home country code
```

XML-RPC (PHP structure)

```
[country] => string
```

REST (POST parameters)

```
country= string
```



recaptchachallengehash (Default to "")

Recaptcha challenge hash

General structure

```
string Default to "" //Recaptcha challenge hash
```

XML-RPC (PHP structure)

```
[recaptchachallengehash] => string
```

REST (POST parameters)

```
recaptchachallengehash= string
```

recaptcharesponse (Default to "")

Recaptcha response

General structure

```
string Default to "" //Recaptcha response
```

XML-RPC (PHP structure)

```
[recaptcharesponse] => string
```

REST (POST parameters)

```
recaptcharesponse= string
```

customprofilefields (Default to "Array ()")

User custom fields (also known as user profile fields)

General structure

```
Default to "Array
(
)
" //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field
name string //The name of the custom field
value string //Custom field value, can be an encoded json if required
}
)
```

XML-RPC (PHP structure)

```
[customprofilefields] =>
Array
(
[0] =>
Array
(
[type] => string
[name] => string
[value] => string
)
)
```

REST (POST parameters)

```
customprofilefields[0][type]= string
customprofilefields[0][name]= string
customprofilefields[0][value]= string
```

**redirect (Default to "")**

Redirect the user to this site url after confirmation.

General structure

```
string Default to "" //Redirect the user to this site url after confirmation.
```

XML-RPC (PHP structure)

```
[redirect] => string
```

REST (POST parameters)

```
redirect= string
```

Response

General structure

```
object {
  success int //True if the user was created false otherwise
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [
    [success] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
          ]
        )
  )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="success">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

block_accessreview_get_module_data

Gets error data for course modules.

Arguments**courseid** (Required)

The course id to obtain results for.

General structure

```
int //The course id to obtain results for.
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
list of (
object {
cmid int //ID
numerrors int //Number of errors.
numchecks int //Number of checks.
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
    [0] =>
        Array
        (
            [cmid] => int
            [numerrors] => int
            [numchecks] => int
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="cmid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="numerrors">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="numchecks">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

block_accessreview_get_section_data

Gets error data for course sections.

Arguments**courseid** (Required)

The course id to obtain results for.

**General structure**

```
int //The course id to obtain results for.
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
list of (
object {
section int //ID
numerrors int //Number of errors.
numchecks int //Number of checks.
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[section] => int
[numerrors] => int
[numchecks] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="section">
<VALUE>int</VALUE>
</KEY>
<KEY name="numerrors">
<VALUE>int</VALUE>
</KEY>
<KEY name="numchecks">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

block_recentlyaccesseditems_get_recent_items

List of items a user has accessed most recently.

Arguments

limit (Default to "0")

result set limit

General structure

```
int Default to "0" //result set limit
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

Response

The most recently accessed activities/resources by the logged user

General structure

```
//The most recently accessed activities/resources by the logged user
list of (
object {
id int //id
courseid int //courseid
cmid int //cmid
userid int //userid
modname string //modname
name string //name
coursename string //coursename
timeaccess int //timeaccess
viewurl string //viewurl
courseviewurl string //courseviewurl
icon string //icon
purpose string //purpose
})
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
    (
        [id] => int
        [courseid] => int
        [cmid] => int
        [userid] => int
        [modname] => string
        [name] => string
        [coursename] => string
        [timeaccess] => int
        [viewurl] => string
        [courseviewurl] => string
        [icon] => string
        [purpose] => string
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="cmid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="modname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="coursename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timeaccess">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="viewurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="courseviewurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="purpose">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

block_starredcourses_get_starred_courses

Get users starred courses.

Arguments

limit (Default to "0")

Limit

General structure

```
int Default to "0" //Limit
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

offset (Default to "0")

Offset

General structure

```
int Default to "0" //Offset
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

Response**General structure**

```
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [fullname] => string
                [shortname] => string
                [idnumber] => string
                [summary] => string
                [summaryformat] => int
                [startdate] => int
                [enddate] => int
                [visible] => int
                [showactivitydates] => int
                [showcompletionconditions] => int
                [pdfexportfont] => string
                [fullnamedisplay] => string
                [viewurl] => string
                [courseimage] => string
                [progress] => int
                [hasprogress] => int
                [isfavourite] => int
                [hidden] => int
                [timeaccess] => int
                [showshortname] => int
                [coursecategory] => string
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summary">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summaryformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="startdate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enddate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showactivitydates">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showcompletionconditions">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="pdfexportfont">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullnamedisplay">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="viewurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="courseimage">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="progress">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hasprogress">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isfavourite">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hidden">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timeaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showshortname">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="coursecategory">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_admin_set_block_protection

Set the protection state for a block plugin

Arguments**plugin** (Required)

The name of the plugin

General structure

```
string //The name of the plugin
```

XML-RPC (PHP structure)

```
[plugin] => string
```

REST (POST parameters)

```
plugin= string
```

state (Required)

The target state

General structure

```
int //The target state
```

**XML-RPC (PHP structure)**

```
[state] => int
```

REST (POST parameters)

```
state= int
```

Response**General structure**

```
object {
}
```

XML-RPC (PHP structure)

```
Array
(
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_admin_set_plugin_order

Set the order of a plugin

Arguments**plugin** (Required)

The name of the plugin

**General structure**

```
string //The name of the plugin
```

XML-RPC (PHP structure)

```
[plugin] => string
```

REST (POST parameters)

```
plugin= string
```

direction (Required)

The direction to move

General structure

```
int //The direction to move
```

XML-RPC (PHP structure)

```
[direction] => int
```

REST (POST parameters)

```
direction= int
```

Response**General structure**

```
object {  
}
```

XML-RPC (PHP structure)

```
Array  
(  
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>  
<RESPONSE>  
    <SINGLE>  
    </SINGLE>  
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>  
<EXCEPTION class="invalid_parameter_exception">  
    <MESSAGE>Invalid parameter value detected</MESSAGE>  
    <DEBUGINFO></DEBUGINFO>  
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_admin_set_plugin_state

Set the state of a plugin

Arguments

plugin (Required)

The name of the plugin

General structure

```
string //The name of the plugin
```

XML-RPC (PHP structure)

```
[plugin] => string
```

REST (POST parameters)

```
plugin= string
```

state (Required)

The target state

General structure

```
int //The target state
```

XML-RPC (PHP structure)

```
[state] => int
```

REST (POST parameters)

```
state= int
```

Response**General structure**

```
object {  
}
```

XML-RPC (PHP structure)

```
Array  
(  
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>  
<RESPONSE>  
    <SINGLE>  
    </SINGLE>  
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>  
<EXCEPTION class="invalid_parameter_exception">  
    <MESSAGE>Invalid parameter value detected</MESSAGE>  
    <DEBUGINFO></DEBUGINFO>  
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_auth_confirm_user

Confirm a user account.

Arguments

username (Required)

User name

General structure

```
string //User name
```

XML-RPC (PHP structure)

```
[username] => string
```

REST (POST parameters)

```
username= string
```

secret (Required)

Confirmation secret

General structure

```
string //Confirmation secret
```

XML-RPC (PHP structure)

```
[secret] => string
```

REST (POST parameters)

```
secret= string
```



Response

General structure

```
object {
  success int //True if the user was confirmed, false if he was already confirmed
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```

Array
(
    [success] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="success">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_auth_is_age_digital_consent_verification_enabled

Checks if age digital consent verification is enabled.

Arguments

Response

General structure

```
object {
    status int //True if digital consent verification is enabled,
               //false otherwise.
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



Restricted to logged-in users

No

Callable from AJAX

Yes

`core_auth_is_minor`

Requests a check if a user is a digital minor.

Arguments

`age` (Required)

Age

General structure

```
int //Age
```

XML-RPC (PHP structure)

```
[age] => int
```

REST (POST parameters)

```
age= int
```

country (Required)

Country of residence

General structure

```
string //Country of residence
```

XML-RPC (PHP structure)

```
[country] => string
```

REST (POST parameters)

```
country= string
```

Response**General structure**

```
object {
    status int //True if the user is considered to be a digital minor,
               //false if not
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

`core_auth_request_password_reset`

Requests a password reset.

Arguments

`username` (Default to "")

User name

General structure

```
string Default to "" //User name
```

XML-RPC (PHP structure)

```
[username] => string
```

REST (POST parameters)

```
username= string
```

`email` (Default to "")

User email

General structure

```
string Default to "" //User email
```

XML-RPC (PHP structure)

```
[email] => string
```

REST (POST parameters)

```
email= string
```

Response

General structure

```

object {
    status string //The returned status of the process:
        dataerror: Error in the sent data (username or email). More information in warnings field.
        emailpasswordconfirmmaybesent: Email sent or not (depends on user found in database).
        emailpasswordconfirmnotsent: Failure, user not found.
        emailpasswordconfirmnoemail: Failure, email not found.
        emailalreadySENT: Email already sent.
        emailpasswordconfirmSENT: User pending confirmation.
        emailresetconfirmSENT: Email sent.

    notice string //Important information for the user about the process.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => string
    [notice] => string
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="notice">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_auth_resend_confirmation_email

Resend confirmation email.

Arguments

username (Required)

Username.

General structure

```
string //Username.
```

XML-RPC (PHP structure)

```
[username] => string
```

REST (POST parameters)

```
username= string
```

password (Required)

Plain text password.



General structure

```
string //Plain text password.
```

XML-RPC (PHP structure)

```
[password] => string
```

REST (POST parameters)

```
password= string
```

redirect (Default to "")

Redirect the user to this site url after confirmation.

General structure

```
string Default to "" //Redirect the user to this site url after confirmation.
```

XML-RPC (PHP structure)

```
[redirect] => string
```

REST (POST parameters)

```
redirect= string
```

Response**General structure**

```
object {
    status int //True if the confirmation email was sent, false otherwise.
    warnings Optional //list of warnings
    list of (
        /warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

**core_backup_get_async_backup_links_backup**

Gets the data to use when updating the status table row in the UI for when an async backup completes.

Arguments**filename** (Required)

Backup filename

General structure

```
string //Backup filename
```

XML-RPC (PHP structure)

```
[filename] => string
```

REST (POST parameters)

```
filename= string
```

contextid (Required)

Context id

General structure

```
int //Context id
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

backupid (Required)

Backup id

General structure

```
string //Backup id
```

XML-RPC (PHP structure)

```
[backupid] => string
```

REST (POST parameters)

```
backupid= string
```

Response

Table row data.

General structure

```
//Table row data.  
object {  
filesize string //Backup file size  
fileurl string //Backup file URL  
restoreurl string //Backup restore URL  
}
```

**XML-RPC (PHP structure)**

```
Array  
(  
[filesize] => string  
[fileurl] => string  
[restoreurl] => string  
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="filesize">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="fileurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="restoreurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_backup_get_async_backup_links_restore

Gets the data to use when updating the status table row in the UI for when an async restore completes.

Arguments**backupid** (Required)

Backup id

**General structure**

```
string //Backup id
```

XML-RPC (PHP structure)

```
[backupid] => string
```

REST (POST parameters)

```
backupid= string
```

contextid (Required)

Context id

General structure

```
int //Context id
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

Response

Table row data.

General structure

```
//Table row data.  
object {  
restoreurl string //Restore url  
}
```

XML-RPC (PHP structure)

```
Array  
(  
[restoreurl] => string  
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>  
<RESPONSE>  
  <SINGLE>  
    <KEY name="restoreurl">  
      <VALUE>string</VALUE>  
    </KEY>  
  </SINGLE>  
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>  
<EXCEPTION class="invalid_parameter_exception">  
  <MESSAGE>Invalid parameter value detected</MESSAGE>  
  <DEBUGINFO></DEBUGINFO>  
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_backup_get_async_backup_progress

Get the progress of an Asyncronous backup.

Arguments**backupids** (Required)

Backup id to get progress for

General structure

```
//Backup id to get progress for
list of (
string //Backup id to get progress for
)
```

XML-RPC (PHP structure)

```
[backupids] =>
Array
(
[0] => string
)
```

REST (POST parameters)

backupids[0]= string

contextid (Required)

Context id

General structure

int //Context id

XML-RPC (PHP structure)

[contextid] => int

REST (POST parameters)

contextid= int

Response

Backup data

General structure

```
//Backup data
list of (
//Backup completion status
object {
status int //Backup Status
progress double //Backup progress
backupid string //Backup id
operation string //operation type
})
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [status] => int
                    [progress] => double
                    [backupid] => string
                    [operation] => string
                )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="status">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="progress">
                <VALUE>double</VALUE>
            </KEY>
            <KEY name="backupid">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="operation">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_backup_get_copy_progress

Gets the progress of course copy operations.

Arguments**copies** (Required)

Copy data

General structure

```
//Copy data
list of (
  //Copy data
object {
  backupid string  //Backup id
  restoreid string //Restore id
  operation string //Operation type
}
)
```

XML-RPC (PHP structure)

```
[copies] =>
  Array
    (
      [0] =>
        Array
          (
            [backupid] => string
            [restoreid] => string
            [operation] => string
          )
    )
)
```

REST (POST parameters)

```
copies[0][backupid]= string
copies[0][restoreid]= string
copies[0][operation]= string
```

Response

Copy data

General structure

```
//Copy data
list of (
  //Copy completion status
object {
  status int //Copy Status
  progress double //Copy progress
  backupid string //Copy id
  operation string //Operation type
}
)
```

**XML-RPC (PHP structure)**

```
Array
  (
    [0] =>
      Array
        (
          [status] => int
          [progress] => double
          [backupid] => string
          [operation] => string
        )
  )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="status">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="progress">
                <VALUE>double</VALUE>
            </KEY>
            <KEY name="backupid">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="operation">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_backup_submit_copy_form

Handles ajax submission of course copy form.

**Arguments****jsonformdata** (Required)

The data from the create copy form, encoded as a json array

General structure

```
string //The data from the create copy form, encoded as a json array
```

XML-RPC (PHP structure)

```
[jsonformdata] => string
```

REST (POST parameters)

```
jsonformdata= string
```

Response

JSON response.

General structure

```
string //JSON response.
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_badges_get_user_badges

Returns the list of badges awarded to a user.

**Arguments****userid** (Default to "0")

Badges only for this user id, empty for current user

General structure

```
int Default to "0" //Badges only for this user id, empty for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

courseid (Default to "0")

Filter badges by course id, empty all the courses

General structure

```
int Default to "0" //Filter badges by course id, empty all the courses
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

page (Default to "0")

The page of records to return.

General structure

```
int Default to "0" //The page of records to return.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "0")

The number of records to return per page

General structure

```
int Default to "0" //The number of records to return per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```



search (Default to "")

A simple string to search for

General structure

```
string Default to "" //A simple string to search for
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

onlypublic (Default to "")

Whether to return only public badges

General structure

```
int Default to "" //Whether to return only public badges
```

XML-RPC (PHP structure)

```
[onlypublic] => int
```

REST (POST parameters)

```
onlypublic= int
```

Response

```

General structure

object {
  badges list of (
    object {
      id int Optional //Badge id
      name string //Badge name
      description string //Badge description
      timecreated int Optional //Time created
      timemodified int Optional //Time modified
      usercreated int Optional //User created
      usermodified int Optional //User modified
      issuername string //Issuer name
      issuerurl string //Issuer URL
      issuercontact string //Issuer contact
      expiredate int Optional //Expire date
      expireperiod int Optional //Expire period
      type int Optional //Type
      courseid int Optional //Course id
      message string Optional //Message
      messagesubject string Optional //Message subject
      attachment int Optional //Attachment
      notification int Optional //Whether to notify when badge is awarded
      nextcron int Optional //Next cron
      status int Optional //Status
      issuedid int Optional //Issued id
      uniquehash string //Unique hash
      dateissued int //Date issued
      dateexpire int //Date expire
      visible int Optional //Visible
      email string Optional //User email
      version string Optional //Version
      language string Optional //Language
      imageauthorname string Optional //Name of the image author
      imageauthoremail string Optional //Email of the image author
      imageauthorurl string Optional //URL of the image author
      imagecaption string Optional //Caption of the image
      badgeurl string //Badge URL
      endorsement object {
        id int //Endorsement id
        badgeid int //Badge id
        issuername string //Endorsement issuer name
        issuerurl string //Endorsement issuer URL
        issueremail string //Endorsement issuer email
        claimid string //Claim URL
        claimcomment string //Claim comment
        dateissued int //Date issued
      }
      alignment //Badge alignments
      list of (
        object {
          id int Optional //Alignment id
          badgeid int Optional //Badge id
          targetName string Optional //Target name
          targetUrl string Optional //Target URL
          targetDescription string Optional //Target description
          targetFramework string Optional //Target framework
          targetCode string Optional //Target code
        }
      )
      relatedbadges //Related badges
      list of (
        object {
          id int //Badge id
          name string //Badge name
          version string Optional //Version
          language string Optional //Language
          type int Optional //Type
        }
      )
    })
  )
  warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}

```



XML-RPC (PHP structure)

```

Array
(
    [badges] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [timecreated] => int
                            [timemodified] => int
                            [usercreated] => int
                            [usermodified] => int
                            [issuername] => string
                            [issuerurl] => string
                            [issuercontact] => string
                            [expiredate] => int
                            [expireperiod] => int
                            [type] => int
                            [courseid] => int
                            [message] => string
                            [messagesubject] => string
                            [attachment] => int
                            [notification] => int
                            [nextcron] => int
                            [status] => int
                            [issuedid] => int
                            [uniquehash] => string
                            [dateissued] => int
                            [dateexpire] => int
                            [visible] => int
                            [email] => string
                            [version] => string
                            [language] => string
                            [imageauthorname] => string
                            [imageauthoremail] => string
                            [imageauthorurl] => string
                            [imagecaption] => string
                            [badgeurl] => string
                            [endorsement] =>
                                Array
                                    (
                                        [id] => int
                                        [badgeid] => int
                                        [issuername] => string
                                        [issuerurl] => string
                                        [issuermail] => string
                                        [claimid] => string
                                        [claimcomment] => string
                                        [dateissued] => int
                                    )
                            [alignment] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [badgeid] => int
                                                    [targetName] => string
                                                    [targetUrl] => string
                                                    [targetDescription] => string
                                                    [targetFramework] => string
                                                    [targetCode] => string
                                                )
                                    )
                            [relatedbadges] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [name] => string
                                                    [version] => string
                                                    [language] => string
                                                    [type] => int
                                                )
                                    )
                        )
                )
)

```



```
        )
    )
[warnings] =>
  Array
  (
    [0] =>
      Array
      (
        [
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
      )
  )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="badges">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usercreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usermodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="issuername">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="issuerurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="issuercontact">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="expiredate">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="expireperiod">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="type">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="messagesubject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="attachment">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="notification">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="nextcron">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="status">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="issuedid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="uniquehash">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="dateissued">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dateexpire">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
```



```

<KEY name="email">
    <VALUE>string</VALUE>
</KEY>
<KEY name="version">
    <VALUE>string</VALUE>
</KEY>
<KEY name="language">
    <VALUE>string</VALUE>
</KEY>
<KEY name="imageauthorname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="imageauthoremail">
    <VALUE>string</VALUE>
</KEY>
<KEY name="imageauthorurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="imagecaption">
    <VALUE>string</VALUE>
</KEY>
<KEY name="badgeurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="endorsement">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="badgeid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="issuernname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="issuerurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="issuermemail">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="claimid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="claimcomment">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="dateissued">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="alignment">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="badgeid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="targetName">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="targetUrl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="targetDescription">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="targetFramework">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="targetCode">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="relatedbadges">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">

```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="version">
<VALUE>string</VALUE>
</KEY>
<KEY name="language">
<VALUE>string</VALUE>
</KEY>
<KEY name="type">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
<SINGLE>
<KEY name="item">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
<VALUE>string</VALUE>
</KEY>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_block_fetch_addable_blocks

Returns all addable blocks in a given page.

Arguments

pagecontextid (Required)

The context ID of the page.

General structure

```
int //The context ID of the page.
```

XML-RPC (PHP structure)

```
[pagecontextid] => int
```

REST (POST parameters)

```
pagecontextid= int
```

pagetype (Required)

The type of the page.

General structure

```
string //The type of the page.
```

XML-RPC (PHP structure)

```
[pagetype] => string
```

REST (POST parameters)

```
pagetype= string
```

pagelayout (Required)

The layout of the page.

General structure

```
string //The Layout of the page.
```

XML-RPC (PHP structure)

```
[pagelayout] => string
```

REST (POST parameters)

```
pagelayout= string
```

**subpage** (Default to "")

The subpage identifier

General structure

```
string Default to "" //The subpage identifier
```

XML-RPC (PHP structure)

```
[subpage] => string
```

REST (POST parameters)

```
subpage= string
```

pagehash (Default to "")

Page hash

General structure

```
string Default to "" //Page hash
```

XML-RPC (PHP structure)

```
[pagehash] => string
```

REST (POST parameters)

```
pagehash= string
```

Response

List of addable blocks in a given page.

General structure

```
//List of addable blocks in a given page.
list of (
object {
name string //The name of the block.
title string //The title of the block.
blockform string //If this block type has a form when it is being added then the classname of the form
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
    (
        [name] => string
        [title] => string
        [blockform] => string
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="title">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="blockform">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_block_get_course_blocks

Returns blocks information for a course.

Arguments**courseid** (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

returncontents (Default to "")

Whether to return the block contents.

General structure

```
int Default to "" //Whether to return the block contents.
```

**XML-RPC (PHP structure)**

```
[returncontents] => int
```

REST (POST parameters)

```
returncontents= int
```

Response

General structure

```
object {
blocks //List of blocks in the course.
list of (
//Block information.
object {
instanceid int //Block instance id.
name string //Block name.
region string //Block region.
positionid int //Position id.
collapsible int //Whether the block is collapsible.
dockable int //Whether the block is dockable.
weight int Optional //Used to order blocks within a region.
visible int Optional //Whether the block is visible.
contents Optional //Block contents (if required).
object {
title string //Block title.
content string //Block contents.
contentformat int //Content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
footer string //Block footer.
files //Block files.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
configs Optional //Block instance and plugin configuration settings.
list of (
object {
name string //Name.
value string //JSON encoded representation of the config value.
type string //Type (instance or plugin).
}
)
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

XML-RPC (PHP structure)

```

Array
(
    [blocks] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [instanceid] => int
                            [name] => string
                            [region] => string
                            [positionid] => int
                            [collapsible] => int
                            [dockable] => int
                            [weight] => int
                            [visible] => int
                            [contents] =>
                                Array
                                    (
                                        [title] => string
                                        [content] => string
                                        [contentformat] => int
                                        [footer] => string
                                        [files] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [filename] => string
                                                                [filepath] => string
                                                                [filesize] => int
                                                                [fileurl] => string
                                                                [timemodified] => int
                                                                [mimetype] => string
                                                                [isexternalfile] => int
                                                                [repositorytype] => string
                                                            )
                                                )
                                    )
                            )
                )
            )
        [configs] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [name] => string
                                [value] => string
                                [type] => string
                            )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="blocks">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="instanceid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="region">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="positionid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="collapsible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dockable">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="weight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="contents">
                        <SINGLE>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="content">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="contentformat">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="footer">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="files">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="filename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="filepath">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="filesize">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="fileurl">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="timemodified">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="mimetype">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="isexternalfile">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="repositorytype">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="configs">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

```

        </KEY>
        <KEY name="value">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="type">
            <VALUE>string</VALUE>
        </KEY>
        <SINGLE>
        </MULTIPLE>
    </KEY>
    <SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes



Callable from AJAX

No

core_block_get_dashboard_blocks

Returns blocks information for the given user dashboard.

Arguments

userid (Default to "0")

User id (optional), default is current user.

General structure

```
int Default to "0" //User id (optional), default is current user.
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

returncontents (Default to "")

Whether to return the block contents.

General structure

```
int Default to "" //Whether to return the block contents.
```

XML-RPC (PHP structure)

```
[returncontents] => int
```

REST (POST parameters)

```
returncontents= int
```

mypage (Default to "__default")

What my page to return blocks of

General structure

```
string Default to "__default" //What my page to return blocks of
```

XML-RPC (PHP structure)

```
[mypage] => string
```

REST (POST parameters)

```
mypage= string
```

Response



General structure

```
object {
blocks //List of blocks in the dashboard.
list of (
//Block information.
object {
instanceid int //Block instance id.
name string //Block name.
region string //Block region.
positionid int //Position id.
collapsible int //Whether the block is collapsible.
dockable int //Whether the block is dockable.
weight int Optional //Used to order blocks within a region.
visible int Optional //Whether the block is visible.
contents Optional //Block contents (if required).
object {
title string //Block title.
content string //Block contents.
contentformat int //Content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
footer string //Block footer.
files //Block files.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
configs Optional //Block instance and plugin configuration settings.
list of (
object {
name string //Name.
value string //JSON encoded representation of the config value.
type string //Type (instance or plugin).
}
)
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [blocks] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [instanceid] => int
                                [name] => string
                                [region] => string
                                [positionid] => int
                                [collapsible] => int
                                [dockable] => int
                                [weight] => int
                                [visible] => int
                                [contents] =>
                                    Array
                                    (
                                        [
                                            [
                                                [title] => string
                                                [content] => string
                                                [contentformat] => int
                                                [footer] => string
                                                [files] =>
                                                    Array
                                                    (
                                                        [
                                                            [
                                                                [0] =>
                                                                    Array
                                                                    (
                                                                        [
                                                                            [
                                                                                [filename] => string
                                                                                [filepath] => string
                                                                                [filesize] => int
                                                                                [fileurl] => string
                                                                                [timemodified] => int
                                                                                [mimetype] => string
                                                                                [isexternalfile] => int
                                                                                [repositorytype] => string
                                                                            )
                                                                        )
                                                                )
                                                        )
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    [configs] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [
                                                [name] => string
                                                [value] => string
                                                [type] => string
                                            )
                                        )
                                    )
                            )
                        )
                    [warnings] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [
                                                [item] => string
                                                [itemid] => int
                                                [warningcode] => string
                                                [message] => string
                                            )
                                        )
                                    )
                            )
                        )
                )
            )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="blocks">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="instanceid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="region">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="positionid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="collapsible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dockable">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="weight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="contents">
                        <SINGLE>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="content">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="contentformat">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="footer">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="files">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="filename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="filepath">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="filesize">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="fileurl">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="timemodified">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="mimetype">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="isexternalfile">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="repositorytype">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="configs">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

```

        </KEY>
        <KEY name="value">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="type">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes



Callable from AJAX

No

core_blog_get_entries

Returns blog entries.

Arguments

filters (Default to "Array ()")

Parameters to filter blog listings.

```
General structure

Default to "Array
(
)
" //Parameters to filter blog listings.
list of (
object {
name string //The expected keys (value format) are:
tag PARAM_NOTAGS blog tag
tagid PARAM_INT blog tag id
userid PARAM_INT blog author (userid)
cmid PARAM_INT course module id
entryid PARAM_INT entry id
groupid PARAM_INT group id
courseid PARAM_INT course id
search PARAM_RAW search term

value string //The value of the filter.
}
)
```

XML-RPC (PHP structure)

```
[filters] =>
  Array
  (
    [0] =>
      Array
      (
        [name] => string
        [value] => string
      )
  )
```

REST (POST parameters)

```
filters[0][name]= string
filters[0][value]= string
```

page (Default to "0")

The blog page to return.

General structure

```
int Default to "0" //The blog page to return.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "10")

The number of posts to return per page.

General structure

```
int Default to "10" //The number of posts to return per page.
```

XML-RPC (PHP structure)

```
[perpage] => int
```

```
REST (POST parameters)
```

```
perpage= int
```

Response

^

General structure

```

object {
entries list of (
object {
id int //Post entry id.
module string //Where it was published the post (blog, blog_external...).
userid int //Post author.
courseid int //Course where the post was created.
groupid int //Group post was created for.
moduleid int //Module id where the post was created (not used anymore).
coursemoduleid int //Course module id where the post was created.
subject string //Post subject.
summary string //Post summary.
summaryformat int Default to "1" //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
content string //Post content.
uniquehash string //Post unique hash.
rating int //Post rating.
format int //Post content format.
attachment string //Post attachment.
publishstate string //Post publish state.
lastmodified int //When it was last modified.
created int //When it was created.
usermodified int //User that updated the post.
summaryfiles //summaryfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)attachmentfiles Optional //attachmentfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)tags Optional //Tags.
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
))
)totalentries int //The total number of entries found.
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
}

```



XML-RPC (PHP structure)

```

Array
(
    [entries] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [module] => string
                            [userid] => int
                            [courseid] => int
                            [groupid] => int
                            [moduleid] => int
                            [coursemoduleid] => int
                            [subject] => string
                            [summary] => string
                            [summaryformat] => int
                            [content] => string
                            [uniquehash] => string
                            [rating] => int
                            [format] => int
                            [attachment] => string
                            [publishstate] => string
                            [lastmodified] => int
                            [created] => int
                            [usermodified] => int
                            [summaryfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [attachmentfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [tags] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [name] => string
                                                    [rawname] => string
                                                    [isstandard] => int
                                                    [tagcollid] => int
                                                    [taginstanceid] => int
                                                    [taginstancecontextid] => int
                                                    [itemid] => int
                                                    [ordering] => int
                                                    [flag] => int
                                                )
                                    )
                            )
                        )
                )
            )
        )
    [totalentries] => int
)

```

```
[warnings] =>
  Array
  (
    [
      [0] =>
        Array
        (
          [
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
          )
        )
    )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="entries">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="module">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="moduleid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursemoduleid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="summary">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="summaryformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="content">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="uniquehash">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rating">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="attachment">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="publishstate">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="lastmodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="created">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usermodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="summaryfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="mimetype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="attachmentfiles">
<MULTIPLE>
    <SINGLE>
        <KEY name="filename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="filepath">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="filesize">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fileurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mimetype">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="isexternalfile">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="repositorytype">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
<MULTIPLE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="rawname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="isstandard">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="tagcollid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="taginstanceid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="taginstancecontextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="ordering">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="flag">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="totalentries">
    <VALUE>int</VALUE>
</KEY>
```



```

</KEY>
<KEY name="warnings">
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_blog_view_entries

Trigger the blog_entries_viewed event.



Arguments

filters (Default to "Array ()")

Parameters used in the filter of view_entries.

```
General structure

Default to "Array
(
)
" //Parameters used in the filter of view_entries.
list of (
object {
name string //The expected keys (value format) are:
tag PARAM_NOTAGS blog tag
tagid PARAM_INT blog tag id
userid PARAM_INT blog author (userid)
cmid PARAM_INT course module id
entryid PARAM_INT entry id
groupid PARAM_INT group id
courseid PARAM_INT course id
search PARAM_RAW search term

value string //The value of the filter.
}
)
```

XML-RPC (PHP structure)

```
[filters] =>
  Array
  (
    [0] =>
      Array
      (
        [name] => string
        [value] => string
      )
  )
```

REST (POST parameters)

```
filters[0][name]= string
filters[0][value]= string
```

Response

```
General structure

object {
status int //status: true if success
warnings Optional //list of warnings
list of (
  /warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
            )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

`core_calendar_create_calendar_events`

Create calendar events

Arguments

events (Required)

General structure

```

list of (
  //event
object {
  name string  //event name
  description string Default to "null" //Description
  format int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
  courseid int Default to "0" //course id
  groupid int Default to "0" //group id
  repeats int Default to "0" //number of repeats
  eventtype string Default to "user" //Event type
  timestart int Default to "1697100132" //timestart
  timeduration int Default to "0" //time duration
  visible int Default to "1" //visible
  sequence int Default to "1" //sequence
}
)

```

XML-RPC (PHP structure)

```

[events] =>
  Array
    (
      [0] =>
        Array
          (
            [name] => string
            [description] => string
            [format] => int
            [courseid] => int
            [groupid] => int
            [repeats] => int
            [eventtype] => string
            [timestart] => int
            [timeduration] => int
            [visible] => int
            [sequence] => int
          )
    )
)

```

REST (POST parameters)

```

events[0][name]= string
events[0][description]= string
events[0][format]= int
events[0][courseid]= int
events[0][groupid]= int
events[0][repeats]= int
events[0][eventtype]= string
events[0][timestart]= int
events[0][timeduration]= int
events[0][visible]= int
events[0][sequence]= int

```



Response

General structure

```
object {
events list of (
    //event
object {
id int //event id
name string //event name
description string Optional //Description
format int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
courseid int //course id
groupid int //group id
userid int //user id
repeatid int Optional //repeat id
modulename string Optional //module name
instance int //instance id
eventtype string //Event type
timestart int //timestart
timeduration int //time duration
visible int //visible
uuid string Optional //unique id of ical events
sequence int //sequence
timemodified int //time modified
subscriptionid int Optional //Subscription id
}
)warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```
Array
(
    [events] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [format] => int
                            [courseid] => int
                            [groupid] => int
                            [userid] => int
                            [repeatid] => int
                            [modulename] => string
                            [instance] => int
                            [eventtype] => string
                            [timestart] => int
                            [timeduration] => int
                            [visible] => int
                            [uuid] => string
                            [sequence] => int
                            [timemodified] => int
                            [subscriptionid] => int
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
```



REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="uuid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sequence">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subscriptionid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

```
</SINGLE>
</RESPONSE>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_calendar_delete_calendar_events

Delete calendar events

Arguments

events (Required)

General structure

```
list of (
    //List of events to delete
object {
eventid int   //Event ID
repeat int   //Delete complete series if repeated event
}
)
```

XML-RPC (PHP structure)

```
[events] =>
    Array
        (
            [0] =>
                Array
                    (
                        [eventid] => int
                        [repeat] => int
                    )
        )
)
```

**REST (POST parameters)**

```
events[0][eventid]= int
events[0][repeat]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_delete_subscription

Delete the calendar subscription

Arguments**subscriptionid** (Required)

The id of the subscription

General structure

```
int //The id of the subscription
```

XML-RPC (PHP structure)

```
[subscriptionid] => int
```

REST (POST parameters)

```
subscriptionid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
  (
    [status] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_calendar_get_action_events_by_course`

Get calendar action events by course

Arguments

courseid (Required)

Course id

General structure

```
int //Course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

timesortfrom (Default to "null")

Time sort from

General structure

```
int Default to "null" //Time sort from
```

XML-RPC (PHP structure)

```
[timesortfrom] => int
```

REST (POST parameters)

```
timesortfrom= int
```

timesortto (Default to "null")

Time sort to

General structure

```
int Default to "null" //Time sort to
```

XML-RPC (PHP structure)

```
[timesortto] => int
```

**REST (POST parameters)**

```
timesortto= int
```

aftereventid (Default to "0")

The last seen event id

General structure

```
int Default to "0" //The last seen event id
```

XML-RPC (PHP structure)

```
[aftereventid] => int
```

REST (POST parameters)`aftereventid= int`**limitnum (Default to "20")**

Limit number

General structure`int Default to "20" //Limit number`**XML-RPC (PHP structure)**`[limitnum] => int`**REST (POST parameters)**`limitnum= int`**searchvalue (Default to "null")**

The value a user wishes to search against

General structure`string Default to "null" //The value a user wishes to search against`**XML-RPC (PHP structure)**`[searchvalue] => string`**REST (POST parameters)**`searchvalue= string`**Response**

```

General structure

object {
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //Location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
}

```



```
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
)
firstid int //firstid
lastid int //lastid
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [events] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [descriptionformat] => int
                            [location] => string
                            [categoryid] => int
                            [groupid] => int
                            [userid] => int
                            [repeatid] => int
                            [eventcount] => int
                            [component] => string
                            [modulename] => string
                            [activityname] => string
                            [activitystr] => string
                            [instance] => int
                            [eventtype] => string
                            [timestart] => int
                            [timeduration] => int
                            [timesort] => int
                            [timeusermidnight] => int
                            [visible] => int
                            [timemodified] => int
                            [overdue] => int
                            [icon] =>
                                Array
                                    (
                                        [key] => string
                                        [component] => string
                                        [alttext] => string
                                        [iconurl] => string
                                        [iconclass] => string
                                    )
                            [category] =>
                                Array
                                    (
                                        [id] => int
                                        [name] => string
                                        [idnumber] => string
                                        [description] => string
                                        [parent] => int
                                        [coursecount] => int
                                        [visible] => int
                                        [timemodified] => int
                                        [depth] => int
                                        [nestedname] => string
                                        [url] => string
                                    )
                            [course] =>
                                Array
                                    (
                                        [id] => int
                                        [fullname] => string
                                        [shortname] => string
                                        [idnumber] => string
                                        [summary] => string
                                        [summaryformat] => int
                                        [startdate] => int
                                        [enddate] => int
                                        [visible] => int
                                        [showactivitydates] => int
                                        [showcompletionconditions] => int
                                        [pdfexportfont] => string
                                        [fullnamedisplay] => string
                                        [viewurl] => string
                                        [courseimage] => string
                                        [progress] => int
                                        [hasprogress] => int
                                        [isfavourite] => int
                                        [hidden] => int
                                        [timeaccess] => int
                                        [showshortname] => int
                                        [coursecategory] => string
                                    )
                        )
                )
            )
)

```



```
        )
[subscription] =>
  Array
    (
      [displayeventsourcename] => int
      [subscriptionname] => string
      [subscriptionurl] => string
    )
[canedit] => int
[candelete] => int
[deleteurl] => string
[editurl] => string
[viewurl] => string
[formattedtime] => string
[formatedlocation] => string
[isactionevent] => int
[iscourseevent] => int
[iscategoryevent] => int
[groupname] => string
[normalisedeventtype] => string
[normalisedeventtypeext] => string
[action] =>
  Array
    (
      [name] => string
      [url] => string
      [itemcount] => int
      [actionable] => int
      [showitemcount] => int
    )
[purpose] => string
[url] => string
)
)
[firstid] => int
[lastid] => int
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="location">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activityname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activitystr">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timesort">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeusermidnight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="overdue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsourc">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
```



```

<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="firstid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lastid">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_get_action_events_by_courses

Get calendar action events by courses

Arguments

courseids (Required)

General structure

```
list of (
int //Course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
courseids[0]= int
```

timesortfrom (Default to "null")

Time sort from

General structure

```
int Default to "null" //Time sort from
```

XML-RPC (PHP structure)

```
[timesortfrom] => int
```

REST (POST parameters)

```
timesortfrom= int
```

timesortto (Default to "null")

Time sort to

General structure

```
int Default to "null" //Time sort to
```

**XML-RPC (PHP structure)**

```
[timesortto] => int
```

REST (POST parameters)

```
timesortto= int
```

limitnum (Default to "10")

Limit number

General structure

```
int Default to "10" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)`limitnum= int`**searchvalue** (Default to "null")

The value a user wishes to search against

General structure`string Default to "null" //The value a user wishes to search against`**XML-RPC (PHP structure)**`[searchvalue] => string`**REST (POST parameters)**`searchvalue= string`**Response**

```

General structure

object {
groupedbycourse //groupedbycourse
list of (
object {
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
}

```



```
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
}
firstid int //firstid
lastid int //lastid
courseid int //courseid
}
)}
```



XML-RPC (PHP structure)

```

Array
(
    [groupedbycourse] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [events] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [
                                                       id] => int
                                                       name] => string
                                                       description] => string
                                                       descriptionformat] => int
                                                       location] => string
                                                       categoryid] => int
                                                       groupid] => int
                                                       userid] => int
                                                       repeatid] => int
                                                       eventcount] => int
                                                       component] => string
                                                       modulename] => string
                                                       activityname] => string
                                                       activitystr] => string
                                                       instance] => int
                                                       eventtype] => string
                                                       timestamp] => int
                                                       timeduration] => int
                                                       timesort] => int
                                                       timeusermidnight] => int
                                                       visible] => int
                                                       timemodified] => int
                                                       overdue] => int
                                                       icon] =>
                                                       Array
                                                       (
                                                       [key] => string
                                                       component] => string
                                                       alttext] => string
                                                       iconurl] => string
                                                       iconclass] => string
                                                       )
                                            )
                                            [category] =>
                                                Array
                                                    (
                                                       [id] => int
                                                       name] => string
                                                       idnumber] => string
                                                       description] => string
                                                       parent] => int
                                                       coursecount] => int
                                                       visible] => int
                                                       timemodified] => int
                                                       depth] => int
                                                       nestedname] => string
                                                       url] => string
                                                       )
                                            )
                                            [course] =>
                                                Array
                                                    (
                                                       [id] => int
                                                       fullname] => string
                                                       shortname] => string
                                                       idnumber] => string
                                                       summary] => string
                                                       summaryformat] => int
                                                       startdate] => int
                                                       enddate] => int
                                                       visible] => int
                                                       showactivitydates] => int
                                                       showcompletionconditions] => int
                                                       pdfexportfont] => string
                                                       fullnamedisplay] => string
                                                       viewurl] => string
                                                       courseimage] => string
                                                       progress] => int
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
)

```





REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="groupedbycourse">
      <MULTIPLE>
        <SINGLE>
          <KEY name="events">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="description">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="descriptionformat">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="location">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="categoryid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="groupid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="repeatid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="eventcount">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="component">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="modulename">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="activityname">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="activitystr">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="instance">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="eventtype">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="timestart">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="timeduration">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="timesort">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="timeusermidnight">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="visible">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="overdue">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="icon">
                  <SINGLE>
                    <KEY name="key">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="component">
    <VALUE>string</VALUE>
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsources">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candele">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
```



```
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypepetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="firstid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lastid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_calendar_get_action_events_by_timesort`

Get calendar action events by tiemsort

Arguments

timesortfrom (Default to "0")

Time sort from

General structure

```
int Default to "0" //Time sort from
```

XML-RPC (PHP structure)

```
[timesortfrom] => int
```

REST (POST parameters)

```
timesortfrom= int
```

timesortto (Default to "null")

Time sort to

General structure

```
int Default to "null" //Time sort to
```

XML-RPC (PHP structure)

```
[timesortto] => int
```

REST (POST parameters)

```
timesortto= int
```

aftereventid (Default to "0")

The last seen event id

General structure

```
int Default to "0" //The last seen event id
```



XML-RPC (PHP structure)

```
[aftereventid] => int
```

REST (POST parameters)

```
aftereventid= int
```

limitnum (Default to "20")

Limit number

General structure

```
int Default to "20" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

limitononsuspendedevents (Default to "")

Limit the events to courses the user is not suspended in

General structure

```
int Default to "" //Limit the events to courses the user is not suspended in
```

XML-RPC (PHP structure)

```
[limitononsuspendedevents] => int
```

REST (POST parameters)

```
limitononsuspendedevents= int
```

userid (Default to "null")

The userid

General structure

```
int Default to "null" //The user id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

searchvalue (Default to "null")

The value a user wishes to search against

General structure

```
string Default to "null" //The value a user wishes to search against
```

**XML-RPC (PHP structure)**

```
[searchvalue] => string
```

REST (POST parameters)

```
searchvalue= string
```

Response

```

General structure

object {
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //Location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
}

```



```
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
)
firstid int //firstid
lastid int //lastid
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [events] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [descriptionformat] => int
                            [location] => string
                            [categoryid] => int
                            [groupid] => int
                            [userid] => int
                            [repeatid] => int
                            [eventcount] => int
                            [component] => string
                            [modulename] => string
                            [activityname] => string
                            [activitystr] => string
                            [instance] => int
                            [eventtype] => string
                            [timestart] => int
                            [timeduration] => int
                            [timesort] => int
                            [timeusermidnight] => int
                            [visible] => int
                            [timemodified] => int
                            [overdue] => int
                            [icon] =>
                                Array
                                    (
                                        [key] => string
                                        [component] => string
                                        [alttext] => string
                                        [iconurl] => string
                                        [iconclass] => string
                                    )
                            [category] =>
                                Array
                                    (
                                        [id] => int
                                        [name] => string
                                        [idnumber] => string
                                        [description] => string
                                        [parent] => int
                                        [coursecount] => int
                                        [visible] => int
                                        [timemodified] => int
                                        [depth] => int
                                        [nestedname] => string
                                        [url] => string
                                    )
                            [course] =>
                                Array
                                    (
                                        [id] => int
                                        [fullname] => string
                                        [shortname] => string
                                        [idnumber] => string
                                        [summary] => string
                                        [summaryformat] => int
                                        [startdate] => int
                                        [enddate] => int
                                        [visible] => int
                                        [showactivitydates] => int
                                        [showcompletionconditions] => int
                                        [pdfexportfont] => string
                                        [fullnamedisplay] => string
                                        [viewurl] => string
                                        [courseimage] => string
                                        [progress] => int
                                        [hasprogress] => int
                                        [isfavourite] => int
                                        [hidden] => int
                                        [timeaccess] => int
                                        [showshortname] => int
                                        [coursecategory] => string
                                    )
                        )
                )
            )
)

```



```
        )
[subscription] =>
  Array
    (
      [displayeventsourcename] => int
      [subscriptionname] => string
      [subscriptionurl] => string
    )
[canedit] => int
[candelete] => int
[deleteurl] => string
[editurl] => string
[viewurl] => string
[formattedtime] => string
[formatedlocation] => string
[isactionevent] => int
[iscourseevent] => int
[iscategoryevent] => int
[groupname] => string
[normalisedeventtype] => string
[normalisedeventtypeext] => string
[action] =>
  Array
    (
      [name] => string
      [url] => string
      [itemcount] => int
      [actionable] => int
      [showitemcount] => int
    )
[purpose] => string
[url] => string
)
)
[firstid] => int
[lastid] => int
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="location">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activityname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activitystr">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timesort">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeusermidnight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="overdue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsourc">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
```



```

<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="firstid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lastid">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_get_allowed_event_types

Get the type of events a user can create in the given course.

Arguments

courseid (Default to "0")

Course to check, empty for site.

General structure

```
int Default to "0" //Course to check, empty for site.
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
    allowedeventtypes list of (
        string //Allowed event types to be created in the given course.
    )warnings Optional //List of warnings
    list of (
        //warning
    )item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
}
```

XML-RPC (PHP structure)

```
Array
(
    [allowedeventtypes] =>
        Array
            (
                [0] => string
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="allowedeventtypes">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_calendar_get_calendar_access_information

Convenience function to retrieve some permissions/access information for the given course calendar.

Arguments**courseid** (Default to "0")

Course to check, empty for site calendar events.

General structure

```
int Default to "0" //Course to check, empty for site calendar events.
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

courseid= int

Response**General structure**

```
object {
    canmanageentries int   //Whether the user can manage entries.
    canmanageownentries int  //Whether the user can manage its own entries.
    canmanagegroupentries int  //Whether the user can manage group entries.
    warnings  Optional //List of warnings
    list of (
        //warning
    object {
        item string  Optional //item
        itemid int  Optional //item id
        warningcode string  //the warning code can be used by the client app to implement specific behaviour
        message string  //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [canmanageentries] => int
    [canmanageownentries] => int
    [canmanagegroupentries] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="canmanageentries">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanageownentries">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanagegroupentries">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

core_calendar_get_calendar_day_view

Fetch the day view data for a calendar

Arguments**year** (Required)

Year to be viewed

General structure

```
int //Year to be viewed
```

XML-RPC (PHP structure)

```
[year] => int
```

REST (POST parameters)

```
year= int
```

month (Required)

Month to be viewed

General structure

```
int //Month to be viewed
```

XML-RPC (PHP structure)

```
[month] => int
```

REST (POST parameters)

```
month= int
```

day (Required)

Day to be viewed

General structure

```
int //Day to be viewed
```

XML-RPC (PHP structure)

```
[day] => int
```

REST (POST parameters)

```
day= int
```

courseid (Default to "1")

Course being viewed

General structure

```
int Default to "1" //Course being viewed
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

categoryid (Default to "null")

Category being viewed

General structure

```
int Default to "null" //Category being viewed
```

XML-RPC (PHP structure)

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

Response



```

General structure

object {
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //Location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
}

```



```
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtype text //normalisedeventtype text
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
islastday int //islastday
popupname string //popupname
mindaytimestamp int Optional //mindaytimestamp
mindayerror string Optional //mindayerror
maxdaytimestamp int Optional //maxdaytimestamp
maxdayerror string Optional //maxdayerror
draggable int //draggable
)
defaulteventcontext int //defaulteventcontext
filter_selector string //filter_selector
courseid int //courseid
categoryid int Optional //categoryid
neweventtimestamp int //neweventtimestamp
date object {
    seconds int //seconds
    minutes int //minutes
    hours int //hours
    mday int //mday
    wday int //wday
    mon int //mon
    year int //year
    yday int //yday
    weekday string //weekday
    month string //month
    timestamp int //timestamp
}
periodname string //periodname
previousperiod object {
    seconds int //seconds
    minutes int //minutes
    hours int //hours
    mday int //mday
    wday int //wday
    mon int //mon
    year int //year
    yday int //yday
    weekday string //weekday
    month string //month
    timestamp int //timestamp
}
previousperiodlink string //previousperiodlink
previousperiodname string //previousperiodname
nextperiod object {
    seconds int //seconds
    minutes int //minutes
    hours int //hours
    mday int //mday
    wday int //wday
    mon int //mon
    year int //year
    yday int //yday
    weekday string //weekday
    month string //month
    timestamp int //timestamp
}
nextperiodname string //nextperiodname
nextperiodlink string //nextperiodlink
larrow string //Larrow
rarrow string //rarrow
}
```



XML-RPC (PHP structure)

```

Array
(
    [events] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [descriptionformat] => int
                            [location] => string
                            [categoryid] => int
                            [groupid] => int
                            [userid] => int
                            [repeatid] => int
                            [eventcount] => int
                            [component] => string
                            [modulename] => string
                            [activityname] => string
                            [activitystr] => string
                            [instance] => int
                            [eventtype] => string
                            [timestart] => int
                            [timeduration] => int
                            [timesort] => int
                            [timeusermidnight] => int
                            [visible] => int
                            [timemodified] => int
                            [overdue] => int
                            [icon] =>
                                Array
                                    (
                                        [key] => string
                                        [component] => string
                                        [alttext] => string
                                        [iconurl] => string
                                        [iconclass] => string
                                    )
                            [category] =>
                                Array
                                    (
                                        [id] => int
                                        [name] => string
                                        [idnumber] => string
                                        [description] => string
                                        [parent] => int
                                        [coursecount] => int
                                        [visible] => int
                                        [timemodified] => int
                                        [depth] => int
                                        [nestedname] => string
                                        [url] => string
                                    )
                            [course] =>
                                Array
                                    (
                                        [id] => int
                                        [fullname] => string
                                        [shortname] => string
                                        [idnumber] => string
                                        [summary] => string
                                        [summaryformat] => int
                                        [startdate] => int
                                        [enddate] => int
                                        [visible] => int
                                        [showactivitydates] => int
                                        [showcompletionconditions] => int
                                        [pdfexportfont] => string
                                        [fullnamedisplay] => string
                                        [viewurl] => string
                                        [courseimage] => string
                                        [progress] => int
                                        [hasprogress] => int
                                        [isfavourite] => int
                                        [hidden] => int
                                        [timeaccess] => int
                                        [showshortname] => int
                                        [coursecategory] => string
                                    )
                        )
                )
            )
)

```



```

        )
    [subscription] =>
        Array
        (
            [displayeventsourcename] => string
            [subscriptionname] => string
            [subscriptionurl] => string
        )
    [canedit] => int
    [candelete] => int
    [deleteurl] => string
    [editurl] => string
    [viewurl] => string
    [formatteddatetime] => string
    [formattedlocation] => string
    [isactionevent] => int
    [iscourseevent] => int
    [iscategoryevent] => int
    [groupname] => string
    [normalisedeventtype] => string
    [normalisedeventtypetext] => string
    [action] =>
        Array
        (
            [name] => string
            [url] => string
            [itemcount] => int
            [actionable] => int
            [showitemcount] => int
        )
    [purpose] => string
    [url] => string
    [islastday] => int
    [popupname] => string
    [mindaytimestamp] => int
    [mindayerror] => string
    [maxdaytimestamp] => int
    [maxdayerror] => string
    [draggable] => int
)
)
[defaulteventcontext] => int
[filter_selector] => string
[courseid] => int
[categoryid] => int
[neweventtimestamp] => int
[date] =>
    Array
    (
        [seconds] => int
        [minutes] => int
        [hours] => int
        [mday] => int
        [wday] => int
        [mon] => int
        [year] => int
        [yday] => int
        [weekday] => string
        [month] => string
        [timestamp] => int
    )
[periodname] => string
[previousperiod] =>
    Array
    (
        [seconds] => int
        [minutes] => int
        [hours] => int
        [mday] => int
        [wday] => int
        [mon] => int
        [year] => int
        [yday] => int
        [weekday] => string
        [month] => string
        [timestamp] => int
    )
[previousperiodlink] => string
[previousperiodname] => string
[nextperiod] =>
    Array
    (
        [seconds] => int
    )

```

```
[minutes] => int
[hours] => int
[mday] => int
[wday] => int
[mon] => int
[year] => int
[yday] => int
[weekday] => string
[month] => string
[timestamp] => int
)
[nextperiodname] => string
[nextperiodlink] => string
[larrow] => string
[rarrow] => string
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="location">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activityname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activitystr">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timesort">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeusermidnight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="overdue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsourc">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
```



```
<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
<KEY name="islastday">
    <VALUE>int</VALUE>
</KEY>
<KEY name="popupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="mindaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mindayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="maxdaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxdayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="draggable">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="defaulteventcontext">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filter_selector">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="categoryid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="neweventtimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="date">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
```



```
</KEY>
<KEY name="wday">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mon">
    <VALUE>int</VALUE>
</KEY>
<KEY name="year">
    <VALUE>int</VALUE>
</KEY>
<KEY name="yday">
    <VALUE>int</VALUE>
</KEY>
<KEY name="weekday">
    <VALUE>string</VALUE>
</KEY>
<KEY name="month">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timestamp">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="periodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="previousperiod">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestamp">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="previousperiodlink">
    <VALUE>string</VALUE>
</KEY>
<KEY name="previousperiodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="nextperiod">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
```



```

        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestamp">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="nextperiodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="nextperiodlink">
    <VALUE>string</VALUE>
</KEY>
<KEY name="larrow">
    <VALUE>string</VALUE>
</KEY>
<KEY name="rarrow">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_calendar_get_calendar_event_by_id`

Get calendar event by id

Arguments

`eventid` (Required)

The event id to be retrieved

General structure

```
int //The event id to be retrieved
```

XML-RPC (PHP structure)

```
[eventid] => int
```

REST (POST parameters)

```
eventid= int
```

Response

```

General structure

object {
event object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
formattedtime string //formattedtime
formattedlocation string //formattedlocation
}

```



```
isactionevent int    //isactionevent
iscourseevent int   //iscourseevent
iscategoryevent int //iscategoryevent
groupname string  Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
  name string //name
  url string //url
  itemcount int //itemcount
  actionable int //actionable
  showitemcount int //showitemcount
}
purpose string //purpose
url string //url
}
warnings Optional //List of warnings
list of (
  //warning
object {
  item string Optional //item
  itemid int Optional //item id
  warningcode string //the warning code can be used by the client app to implement specific behaviour
  message string //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```

Array
(
    [event] =>
        Array
        (
            [id] => int
            [name] => string
            [description] => string
            [descriptionformat] => int
            [location] => string
            [categoryid] => int
            [groupid] => int
            [userid] => int
            [repeatid] => int
            [eventcount] => int
            [component] => string
            [modulename] => string
            [activityname] => string
            [activitystr] => string
            [instance] => int
            [eventtype] => string
            [timestart] => int
            [timeduration] => int
            [timesort] => int
            [timeusermidnight] => int
            [visible] => int
            [timemodified] => int
            [overdue] => int
            [icon] =>
                Array
                (
                    [key] => string
                    [component] => string
                    [alttext] => string
                    [iconurl] => string
                    [iconclass] => string
                )
            [category] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [idnumber] => string
                    [description] => string
                    [parent] => int
                    [coursecount] => int
                    [visible] => int
                    [timemodified] => int
                    [depth] => int
                    [nestedname] => string
                    [url] => string
                )
            [course] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                    [idnumber] => string
                    [summary] => string
                    [summaryformat] => int
                    [startdate] => int
                    [enddate] => int
                    [visible] => int
                    [showactivitydates] => int
                    [showcompletionconditions] => int
                    [pdfexportfont] => string
                    [fullnamedisplay] => string
                    [viewurl] => string
                    [courseimage] => string
                    [progress] => int
                    [hasprogress] => int
                    [isfavourite] => int
                    [hidden] => int
                    [timeaccess] => int
                    [showshortname] => int
                    [coursecategory] => string
                )
            [subscription] =>
                Array

```



```
(  
    [displayeventsource] => int  
    [subscriptionname] => string  
    [subscriptionurl] => string  
)  
[canedit] => int  
[candelete] => int  
[deleteurl] => string  
[editurl] => string  
[viewurl] => string  
[formattedtime] => string  
[formattedlocation] => string  
[isactionevent] => int  
[iscourseevent] => int  
[iscategoryevent] => int  
[groupname] => string  
[normalisedeventtype] => string  
[normalisedeventtypeptext] => string  
[action] =>  
    Array  
        (  
            (  
                [name] => string  
                [url] => string  
                [itemcount] => int  
                [actionable] => int  
                [showitemcount] => int  
            )  
        )  
    [purpose] => string  
    [url] => string  
)  
[warnings] =>  
    Array  
        (  
            [0] =>  
                Array  
                    (  
                        (  
                            [item] => string  
                            [itemid] => int  
                            [warningcode] => string  
                            [message] => string  
                        )  
                    )  
        )  
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="event">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="location">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="categoryid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="repeatid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventcount">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="component">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="modulename">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activityname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activitystr">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="instance">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventtype">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestart">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeduration">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timesort">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeusermidnight">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="overdue">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="icon">
          <SINGLE>
            <KEY name="key">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="component">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </RESPONSE>
```



```
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
            <VALUE>int</VALUE>
        </KEY>
```



```
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsources">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtype">
```



```

        <VALUE>string</VALUE>
    </KEY>
    <KEY name="normalisedeventtypepetext">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="action">
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemcount">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="actionable">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="showitemcount">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </KEY>
    <KEY name="purpose">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="url">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_get_calendar_events

Get calendar events

Arguments

events (Default to "Array ()")

Event details

```

General structure

Default to "Array
(
)
" //Event details
object {
eventids Default to "Array
(
)
" //List of event ids
list of (
int //event ids
)courseids Default to "Array
(
)
" //List of course ids for which events will be returned
list of (
int //course ids
)groupids Default to "Array
(
)
" //List of group ids for which events should be returned
list of (
int //group ids
)categoryids Default to "Array
(
)
" //List of category ids for which events will be returned
list of (
int //Category ids
)}

```

XML-RPC (PHP structure)

```

[events] =>
  Array
    (
      [eventids] =>
        Array
          (
            [0] => int
          )
      [courseids] =>
        Array
          (
            [0] => int
          )
      [groupids] =>
        Array
          (
            [0] => int
          )
      [categoryids] =>
        Array
          (
            [0] => int
          )
    )
  
```



REST (POST parameters)

```

events[eventids][0]= int
events[courseids][0]= int
events[groupids][0]= int
events[categoryids][0]= int

```

options (Default to "Array ())")

Options

```
General structure

Default to "Array
(
)
" //Options
object {
userevents int Default to "1" //Set to true to return current user's user events
siteevents int Default to "1" //Set to true to return site events
timestart int Default to "0" //Time from which events should be returned
timeend int Default to "0" //Time to which the events should be returned. We treat 0 and null as no end
ignorehidden int Default to "1" //Ignore hidden events or not
}
```

XML-RPC (PHP structure)

```
[options] =>
  Array
  (
    [userevents] => int
    [siteevents] => int
    [timestart] => int
    [timeend] => int
    [ignorehidden] => int
  )
```

REST (POST parameters)

```
options[userevents]= int
options[siteevents]= int
options[timestart]= int
options[timeend]= int
options[ignorehidden]= int
```

Response



General structure

```
object {
events list of (
    //event
object {
id int //event id
name string //event name
description string Optional //Description
format int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
courseid int //course id
categoryid int Optional //Category id (only for category events).
groupid int //group id
userid int //user id
repeatid int //repeat id
modulename string Optional //module name
instance int //instance id
eventtype string //Event type
timestart int //timestart
timeduration int //time duration
visible int //visible
uuid string Optional //unique id of ical events
sequence int //sequence
timemodified int //time modified
subscriptionid int optional //Subscription id
}
)warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```
Array
(
    [events] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                            [format] => int
                            [courseid] => int
                            [categoryid] => int
                            [groupid] => int
                            [userid] => int
                            [repeatid] => int
                            [modulename] => string
                            [instance] => int
                            [eventtype] => string
                            [timestart] => int
                            [timeduration] => int
                            [visible] => int
                            [uuid] => string
                            [sequence] => int
                            [timemodified] => int
                            [subscriptionid] => int
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="uuid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sequence">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subscriptionid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </RESPONSE>
```

```
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_calendar_get_calendar_export_token

Return the auth token required for exporting a calendar.

Arguments

Response

General structure

```
object {
    token string //The calendar permanent access token for calendar export.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
    }
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
}
```



XML-RPC (PHP structure)

```
Array
(
    [token] => string
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="token">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`core_calendar_get_calendar_monthly_view`

Fetch the monthly view data for a calendar

Arguments

year (Required)

Year to be viewed

General structure

```
int //Year to be viewed
```

XML-RPC (PHP structure)

```
[year] => int
```

REST (POST parameters)

```
year= int
```

month (Required)

Month to be viewed

General structure

```
int //Month to be viewed
```

XML-RPC (PHP structure)

```
[month] => int
```

REST (POST parameters)

```
month= int
```

courseid (Default to "1")

Course being viewed

General structure

```
int Default to "1" //Course being viewed
```

XML-RPC (PHP structure)

```
[courseid] => int
```



REST (POST parameters)

```
courseid= int
```

categoryid (Default to "null")

Category being viewed

General structure

```
int Default to "null" //Category being viewed
```

XML-RPC (PHP structure)

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

includenavigation (Default to "1")

Whether to show course navigation

General structure

```
int Default to "1" //Whether to show course navigation
```

XML-RPC (PHP structure)

```
[includenavigation] => int
```

REST (POST parameters)

```
includenavigation= int
```

mini (Default to "")

Whether to return the mini month view or not

General structure

```
int Default to "" //Whether to return the mini month view or not
```

XML-RPC (PHP structure)

```
[mini] => int
```

REST (POST parameters)

```
mini= int
```

day (Default to "1")

Day to be viewed

General structure

```
int Default to "1" //Day to be viewed
```

**XML-RPC (PHP structure)**

```
[day] => int
```

REST (POST parameters)

```
day= int
```

view (Default to "month")

The view mode of the calendar

General structure

```
string Default to "month" //The view mode of the calendar
```

XML-RPC (PHP structure)

```
[view] => string
```

```
REST (POST parameters)
```

```
view= string
```

Response

^

```
General structure

object {
url string //url
courseid int //courseid
categoryid int Optional //categoryid
filter_selector string Optional //filter_selector
weeks //weeks
list of (
object {
prepadding //prepadding
list of (
int //prepadding
)postpadding //postpadding
list of (
int //postpadding
)days //days
list of (
object {
seconds int //seconds
minutes int //minutes
hours int //hours
mday int //mday
wday int //wday
year int //year
yday int //yday
istoday int //istoday
isweekend int //isweekend
timestamp int //timestamp
neweventtimestamp int //neweventtimestamp
viewdaylink string Optional //viewdayLink
viewdaylinktitle string Optional //viewdayLinktitle
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
location string Optional //location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
}
```

```

summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsources int //displayeventsources
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtype text string //normalisedeventtype text
action object {
name string //name
url string //url
itemcount int //itemcount
actionable int //actionable
showitemcount int //showitemcount
}
purpose string //purpose
url string //url
islastday int //islastday
popupname string //popupname
mindaytimestamp int Optional //mindaytimestamp
mindayerror string Optional //mindayerror
maxdaytimestamp int Optional //maxdaytimestamp
maxdayerror string Optional //maxdayerror
draggable int //draggable
}
)hasevents int //hasevents
calendareventtypes //calendareventtypes
list of (
string //calendareventtypes
)previousperiod int //previousperiod
nextperiod int //nextperiod
navigation string //navigation
haslastdayofevent int //haslastdayofevent
popovertitle string //popovertitle
daytitle string //daytitle
)
)
)daynames //daynames
list of (
object {
dayno int //dayno
shortname string //shortname
fullname string //fullname
}
)view string //view
date object {
seconds int //seconds
minutes int //minutes
hours int //hours
mday int //mday
wday int //wday
mon int //mon
year int //year
yday int //yday
}

```



```
weekday string //weekday
month string //month
timestamp int //timestamp
}
periodname string //periodname
includenavigation int //includenavigation
initialeventsloaded int //initialeventsloaded
previousperiod object {
seconds int //seconds
minutes int //minutes
hours int //hours
mday int //mday
wday int //wday
mon int //mon
year int //year
yday int //yday
weekday string //weekday
month string //month
timestamp int //timestamp
}
previousperiodlink string //previousperiodlink
previousperiodname string //previousperiodname
nextperiod object {
seconds int //seconds
minutes int //minutes
hours int //hours
mday int //mday
wday int //wday
mon int //mon
year int //year
yday int //yday
weekday string //weekday
month string //month
timestamp int //timestamp
}
nextperiodname string //nextperiodname
nextperiodlink string //nextperiodlink
larrow string //larrow
rarrow string //rarrow
defaulteventcontext int //defaulteventcontext
calendarinstanceid int //calendarinstanceid
viewingmonth int //viewingmonth
showviewselector int //showviewselector
viewinginblock int //viewinginblock
}
```



XML-RPC (PHP structure)

```

Array
(
[url] => string
[courseid] => int
[categoryid] => int
[filter_selector] => string
[weeks] =>
    Array
    (
        [0] =>
            Array
            (
                [prepadding] =>
                    Array
                    (
                        [0] => int
                    )
                [postpadding] =>
                    Array
                    (
                        [0] => int
                    )
            )
        [days] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [seconds] => int
                        [minutes] => int
                        [hours] => int
                        [mday] => int
                        [wday] => int
                        [year] => int
                        [yday] => int
                        [istoday] => int
                        [isweekend] => int
                        [timestamp] => int
                        [neweventtimestamp] => int
                        [viewdaylink] => string
                        [viewdaylinktitle] => string
                    )
                [events] =>
                    Array
                    (
                        [0] =>
                            Array
                            (
                                [id] => int
                                [name] => string
                                [description] => string
                                [descriptionformat] => int
                                [location] => string
                                [categoryid] => int
                                [groupid] => int
                                [userid] => int
                                [repeatid] => int
                                [eventcount] => int
                                [component] => string
                                [modulename] => string
                                [activityname] => string
                                [activitystr] => string
                                [instance] => int
                                [eventtype] => string
                                [timestart] => int
                                [timeduration] => int
                                [timesort] => int
                                [timeusermidnight] => int
                                [visible] => int
                                [timemodified] => int
                                [overdue] => int
                                [icon] =>
                                    Array
                                    (
                                        [key] => string
                                        [component] => string
                                        [alttext] => string
                                        [iconurl] => string
                                        [iconclass] => string
                                    )
                            )
                        [category] =>
                    )
                )
            )
        )
    )
)

```



```

        Array
        (
            [id] => int
            [name] => string
            [idnumber] => string
            [description] => string
            [parent] => int
            [coursecount] => int
            [visible] => int
            [timemodified] => int
            [depth] => int
            [nestedname] => string
            [url] => string
        )
    )
    [course] =>
        Array
        (
            [id] => int
            [fullname] => string
            [shortname] => string
            [idnumber] => string
            [summary] => string
            [summaryformat] => int
            [startdate] => int
            [enddate] => int
            [visible] => int
            [showactivitydates] => int
            [showcompletionconditions] => int
            [pdfexportfont] => string
            [fullnamedisplay] => string
            [viewurl] => string
            [courseimage] => string
            [progress] => int
            [hasprogress] => int
            [isfavourite] => int
            [hidden] => int
            [timeaccess] => int
            [showshortname] => int
            [coursecategory] => string
        )
    )
    [subscription] =>
        Array
        (
            [displayeventsource] => int
            [subscriptionname] => string
            [subscriptionurl] => string
        )
    )
    [canedit] => int
    [candelete] => int
    [deleteurl] => string
    [editurl] => string
    [viewurl] => string
    [formattedtime] => string
    [formattedlocation] => string
    [isactionevent] => int
    [iscourseevent] => int
    [iscategoryevent] => int
    [groupname] => string
    [normalisedeventtype] => string
    [normalisedeventtype] => string
    [action] =>
        Array
        (
            [name] => string
            [url] => string
            [itemcount] => int
            [actionable] => int
            [showitemcount] => int
        )
    )
    [purpose] => string
    [url] => string
    [islastday] => int
    [popupname] => string
    [mindaytimestamp] => int
    [mindayerror] => string
    [maxdaytimestamp] => int
    [maxdayerror] => string
    [draggable] => int
)
)
)
[hasevents] => int
[calendareventtypes] =>
    Array

```



```

        (
        [0] => string
        )
    [previousperiod] => int
    [nextperiod] => int
    [navigation] => string
    [haslastdayofevent] => int
    [popovertitle] => string
    [daytitle] => string
    )
)
)
)
[daynames] =>
Array
(
[0] =>
    Array
    (
        [
            [dayno] => int
            [shortname] => string
            [fullname] => string
        )
    )
)
[view] => string
[date] =>
    Array
    (
        [
            [seconds] => int
            [minutes] => int
            [hours] => int
            [mday] => int
            [wday] => int
            [mon] => int
            [year] => int
            [yday] => int
            [weekday] => string
            [month] => string
            [timestamp] => int
        )
    )
[periodname] => string
[includenavigation] => int
[initialeventsloaded] => int
[previousperiod] =>
    Array
    (
        [
            [seconds] => int
            [minutes] => int
            [hours] => int
            [mday] => int
            [wday] => int
            [mon] => int
            [year] => int
            [yday] => int
            [weekday] => string
            [month] => string
            [timestamp] => int
        )
    )
[previousperiodlink] => string
[previousperiodname] => string
[nextperiod] =>
    Array
    (
        [
            [seconds] => int
            [minutes] => int
            [hours] => int
            [mday] => int
            [wday] => int
            [mon] => int
            [year] => int
            [yday] => int
            [weekday] => string
            [month] => string
            [timestamp] => int
        )
    )
[nextperiodname] => string
[nextperiodlink] => string
[larrow] => string
[rarrow] => string
[defaulteventcontext] => int
[calendarinstanceid] => int
[viewingmonth] => int
[showviewselector] => int

```



```
[viewinginblock] => int  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="courseid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="categoryid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="filter_selector">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="weeks">
      <MULTIPLE>
        <SINGLE>
          <KEY name="prepadding">
            <MULTIPLE>
              <VALUE>int</VALUE>
            </MULTIPLE>
          </KEY>
          <KEY name="postpadding">
            <MULTIPLE>
              <VALUE>int</VALUE>
            </MULTIPLE>
          </KEY>
        <KEY name="days">
          <MULTIPLE>
            <SINGLE>
              <KEY name="seconds">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="minutes">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="hours">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="mday">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="wday">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="year">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="yday">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="istoday">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="isweekend">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="timestamp">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="neweventtimestamp">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="viewdaylink">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="viewdaylinktitle">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="events">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="id">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                      <VALUE>string</VALUE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </MULTIPLE>
          </KEY>
        </MULTIPLE>
      </SINGLE>
    </KEY>
  </MULTIPLE>
</RESPONSE>
```



```
</KEY>
<KEY name="descriptionformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="location">
    <VALUE>string</VALUE>
</KEY>
<KEY name="categoryid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repeatid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="eventcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="component">
    <VALUE>string</VALUE>
</KEY>
<KEY name="modulename">
    <VALUE>string</VALUE>
</KEY>
<KEY name="activityname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="activitystr">
    <VALUE>string</VALUE>
</KEY>
<KEY name="instance">
    <VALUE>int</VALUE>
</KEY>
<KEY name="eventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timestart">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeduration">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timesort">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeusermidnight">
    <VALUE>int</VALUE>
</KEY>
<KEY name="visible">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="overdue">
    <VALUE>int</VALUE>
</KEY>
<KEY name="icon">
    <SINGLE>
        <KEY name="key">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="alttext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="iconurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="iconclass">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
    <VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
    <VALUE>string</VALUE>
</KEY>
<KEY name="description">
    <VALUE>string</VALUE>
</KEY>
<KEY name="parent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="visible">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="depth">
    <VALUE>int</VALUE>
</KEY>
<KEY name="nestedname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="pdfexportfont">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullnamedisplay">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="courseimage">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="progress">
            <VALUE>int</VALUE>
        </KEY>
```



```

</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsourcename">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypepetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>

```



```
</KEY>
<KEY name="actionable">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showitemcount">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
<KEY name="islastday">
    <VALUE>int</VALUE>
</KEY>
<KEY name="popupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="mindaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mindayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="maxdaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxdayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="draggable">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="hasevents">
    <VALUE>int</VALUE>
</KEY>
<KEY name="calendareventtypes">
    <MULTIPLE>
        <VALUE>string</VALUE>
    </MULTIPLE>
</KEY>
<KEY name="previousperiod">
    <VALUE>int</VALUE>
</KEY>
<KEY name="nextperiod">
    <VALUE>int</VALUE>
</KEY>
<KEY name="navigation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="haslastdayofevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="popovertitle">
    <VALUE>string</VALUE>
</KEY>
<KEY name="daytitle">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="daynames">
    <MULTIPLE>
        <SINGLE>
            <KEY name="dayno">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="view">
    <VALUE>string</VALUE>
</KEY>
<KEY name="date">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestamp">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="periodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="includenavigation">
    <VALUE>int</VALUE>
</KEY>
<KEY name="initialeventsloaded">
    <VALUE>int</VALUE>
</KEY>
<KEY name="previousperiod">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
```

^

```
</KEY>
<KEY name="timestamp">
    <VALUE>int</VALUE>
</KEY>
<SINGLE>
</KEY>
<KEY name="previousperiodlink">
    <VALUE>string</VALUE>
</KEY>
<KEY name="previousperiodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="nextperiod">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestamp">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="nextperiodname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="nextperiodlink">
    <VALUE>string</VALUE>
</KEY>
<KEY name="larrow">
    <VALUE>string</VALUE>
</KEY>
<KEY name="rarrow">
    <VALUE>string</VALUE>
</KEY>
<KEY name="defaulteventcontext">
    <VALUE>int</VALUE>
</KEY>
<KEY name="calendarinstanceid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="viewingmonth">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showviewselector">
    <VALUE>int</VALUE>
</KEY>
<KEY name="viewinginblock">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_get_calendar_upcoming_view

Fetch the upcoming view data for a calendar

Arguments**courseid** (Default to "1")

Course being viewed

General structure

```
int Default to "1" //Course being viewed
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

categoryid (Default to "null")

Category being viewed

**General structure**

```
int Default to "null" //Category being viewed
```

XML-RPC (PHP structure)

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

Response

```

General structure

object {
events //events
list of (
object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //Location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsouce int //displayeventsouce
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
}

```



```
formattedtime string //formattedtime
formattedlocation string //formattedlocation
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtype text string //normalisedeventtype text
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
islastday int //islastday
popupname string //popupname
mindaytimestamp int Optional //mindaytimestamp
mindayerror string Optional //mindayerror
maxdaytimestamp int Optional //maxdaytimestamp
maxdayerror string Optional //maxdayerror
draggable int //draggable
)
defaulteventcontext int //defaulteventcontext
filter_selector string //filter_selector
courseid int //courseid
categoryid int Optional //categoryid
isloggedin int //isloggedin
date object {
    seconds int //seconds
    minutes int //minutes
    hours int //hours
    mday int //mday
    wday int //wday
    mon int //mon
    year int //year
    yday int //yday
    weekday string //weekday
    month string //month
    timestamp int //timestamp
}
```



XML-RPC (PHP structure)

```

Array
(
    [events] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [description] => string
                    [descriptionformat] => int
                    [location] => string
                    [categoryid] => int
                    [groupid] => int
                    [userid] => int
                    [repeatid] => int
                    [eventcount] => int
                    [component] => string
                    [modulename] => string
                    [activityname] => string
                    [activitystr] => string
                    [instance] => int
                    [eventtype] => string
                    [timestart] => int
                    [timeduration] => int
                    [timesort] => int
                    [timeusermidnight] => int
                    [visible] => int
                    [timemodified] => int
                    [overdue] => int
                    [icon] =>
                        Array
                        (
                            [key] => string
                            [component] => string
                            [alttext] => string
                            [iconurl] => string
                            [iconclass] => string
                        )
                )
            [category] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [idnumber] => string
                    [description] => string
                    [parent] => int
                    [coursecount] => int
                    [visible] => int
                    [timemodified] => int
                    [depth] => int
                    [nestedname] => string
                    [url] => string
                )
            [course] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                    [idnumber] => string
                    [summary] => string
                    [summaryformat] => int
                    [startdate] => int
                    [enddate] => int
                    [visible] => int
                    [showactivitydates] => int
                    [showcompletionconditions] => int
                    [pdfexportfont] => string
                    [fullnamedisplay] => string
                    [viewurl] => string
                    [courseimage] => string
                    [progress] => int
                    [hasprogress] => int
                    [isfavourite] => int
                    [hidden] => int
                    [timeaccess] => int
                    [showshortname] => int
                    [coursecategory] => string
                )
        )
)

```



```
        )
    [subscription] =>
        Array
            (
                [displayeventsourcename] => int
                [subscriptionname] => string
                [subscriptionurl] => string
            )
    [canedit] => int
    [candelete] => int
    [deleteurl] => string
    [editurl] => string
    [viewurl] => string
    [formattedtime] => string
    [formatedlocation] => string
    [isactionevent] => int
    [iscourseevent] => int
    [iscategoryevent] => int
    [groupname] => string
    [normalisedeventtype] => string
    [normalisedeventtypeext] => string
    [action] =>
        Array
            (
                [name] => string
                [url] => string
                [itemcount] => int
                [actionable] => int
                [showitemcount] => int
            )
    [purpose] => string
    [url] => string
    [islastday] => int
    [popupname] => string
    [mindaytimestamp] => int
    [mindayerror] => string
    [maxdaytimestamp] => int
    [maxdayerror] => string
    [draggable] => int
)
)
[defaulteventcontext] => int
[filter_selector] => string
[courseid] => int
[categoryid] => int
[isloggedin] => int
[date] =>
    Array
        (
            [seconds] => int
            [minutes] => int
            [hours] => int
            [mday] => int
            [wday] => int
            [mon] => int
            [year] => int
            [yday] => int
            [weekday] => string
            [month] => string
            [timestamp] => int
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="events">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="location">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repeatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="modulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activityname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activitystr">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeduration">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timesort">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeusermidnight">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="overdue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsourc">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
```



```
<KEY name="normalisedeventtype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
<KEY name="islastday">
    <VALUE>int</VALUE>
</KEY>
<KEY name="popupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="mindaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mindayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="maxdaytimestamp">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxdayerror">
    <VALUE>string</VALUE>
</KEY>
<KEY name="draggable">
    <VALUE>int</VALUE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="defaultheventcontext">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filter_selector">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="categoryid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isloggedin">
    <VALUE>int</VALUE>
</KEY>
<KEY name="date">
    <SINGLE>
        <KEY name="seconds">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="minutes">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hours">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mday">
            <VALUE>int</VALUE>
```



```

        </KEY>
        <KEY name="wday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="mon">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="year">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="yday">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="weekday">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="month">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestamp">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_get_timestamps



Fetch unix timestamps for given date times.

Arguments

data (Required)

General structure

```

list of (
object {
key string  Optional //key
year int    //year
month int   //month
day int     //day
hour int    Optional //hour
minute int   Optional //minute
})
)

```

XML-RPC (PHP structure)

```
[data] =>
  Array
  (
    [0] =>
      Array
      (
        [key] => string
        [year] => int
        [month] => int
        [day] => int
        [hour] => int
        [minute] => int
      )
  )
```

REST (POST parameters)

```
data[0][key]= string
data[0][year]= int
data[0][month]= int
data[0][day]= int
data[0][hour]= int
data[0][minute]= int
```

Response**General structure**

```
object {
  timestamps list of (
  object {
    key string //Timestamp key
    timestamp int //Unix timestamp
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [timestamps] =>
    Array
    (
      [0] =>
        Array
        (
          [key] => string
          [timestamp] => int
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="timestamps">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="key">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestamp">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_submit_create_update_form

Submit form data for event form

Arguments

formdata (Required)

The data from the event form

**General structure**

string //The data from the event form

XML-RPC (PHP structure)

[formdata] => string

REST (POST parameters)

formdata= string

Response

```

General structure

object {
event object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsources int //displayeventsources
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
formattedtime string //formattedtime
formattedlocation string //formattedlocation
}

```

```
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
}
validationerror int Default to "" //Invalid form data
}
```



XML-RPC (PHP structure)

```

Array
(
    [event] =>
        Array
        (
            [id] => int
            [name] => string
            [description] => string
            [descriptionformat] => int
            [location] => string
            [categoryid] => int
            [groupid] => int
            [userid] => int
            [repeatid] => int
            [eventcount] => int
            [component] => string
            [modulename] => string
            [activityname] => string
            [activitystr] => string
            [instance] => int
            [eventtype] => string
            [timestart] => int
            [timeduration] => int
            [timesort] => int
            [timeusermidnight] => int
            [visible] => int
            [timemodified] => int
            [overdue] => int
            [icon] =>
                Array
                (
                    [key] => string
                    [component] => string
                    [alttext] => string
                    [iconurl] => string
                    [iconclass] => string
                )
            [category] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [idnumber] => string
                    [description] => string
                    [parent] => int
                    [coursecount] => int
                    [visible] => int
                    [timemodified] => int
                    [depth] => int
                    [nestedname] => string
                    [url] => string
                )
            [course] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                    [idnumber] => string
                    [summary] => string
                    [summaryformat] => int
                    [startdate] => int
                    [enddate] => int
                    [visible] => int
                    [showactivitydates] => int
                    [showcompletionconditions] => int
                    [pdfexportfont] => string
                    [fullnamedisplay] => string
                    [viewurl] => string
                    [courseimage] => string
                    [progress] => int
                    [hasprogress] => int
                    [isfavourite] => int
                    [hidden] => int
                    [timeaccess] => int
                    [showshortname] => int
                    [coursecategory] => string
                )
            [subscription] =>
                Array

```



```
(  
    [displayeventsource] => int  
    [subscriptionname] => string  
    [subscriptionurl] => string  
)  
[canedit] => int  
[candelete] => int  
[deleteurl] => string  
[editurl] => string  
[viewurl] => string  
[formattedtime] => string  
[formattedlocation] => string  
[isactionevent] => int  
[iscourseevent] => int  
[iscategoryevent] => int  
[groupname] => string  
[normalisedeventtype] => string  
[normalisedeventtypeptext] => string  
[action] =>  
    Array  
        (  
            (  
                [name] => string  
                [url] => string  
                [itemcount] => int  
                [actionable] => int  
                [showitemcount] => int  
            )  
        )  
    [purpose] => string  
    [url] => string  
)  
[validationerror] => int  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="event">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="location">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="categoryid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="repeatid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventcount">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="component">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="modulename">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activityname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activitystr">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="instance">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventtype">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestart">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeduration">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timesort">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeusermidnight">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="overdue">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="icon">
          <SINGLE>
            <KEY name="key">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="component">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </RESPONSE>
```



```
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
            <VALUE>int</VALUE>
        </KEY>
```



```
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsources">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtype">
```



```

<VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypepetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="validationerror">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_calendar_update_event_start_day

Update the start day (but not time) for an event.

Arguments

eventid (Required)

Id of event to be updated

General structure

```
int //Id of event to be updated
```

XML-RPC (PHP structure)

```
[eventid] => int
```

REST (POST parameters)

```
eventid= int
```

daytimestamp (Required)

Timestamp for the new start day

General structure

```
int //Timestamp for the new start day
```

XML-RPC (PHP structure)

```
[daytimestamp] => int
```

REST (POST parameters)

```
daytimestamp= int
```

Response

```

General structure

object {
event object {
id int //id
name string //name
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
location string Optional //location
categoryid int Optional //categoryid
groupid int Optional //groupid
userid int Optional //userid
repeatid int Optional //repeatid
eventcount int Optional //eventcount
component string Optional //component
modulename string Optional //modulename
activityname string Optional //activityname
activitystr string Optional //activitystr
instance int Optional //instance
eventtype string //eventtype
timestart int //timestart
timeduration int //timeduration
timesort int //timesort
timeusermidnight int //timeusermidnight
visible int //visible
timemodified int //timemodified
overdue int Optional //overdue
icon object {
key string //key
component string //component
alttext string //alttext
iconurl string //iconurl
iconclass string //iconclass
}
category object {
id int //id
name string //name
idnumber string //idnumber
description string Optional //description
parent int //parent
coursecount int //coursecount
visible int //visible
timemodified int //timemodified
depth int //depth
nestedname string //nestedname
url string //url
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
subscription object {
displayeventsources int //displayeventsources
subscriptionname string Optional //subscriptionname
subscriptionurl string Optional //subscriptionurl
}
canedit int //canedit
candelete int //candelete
deleteurl string //deleteurl
editurl string //editurl
viewurl string //viewurl
formattedtime string //formattedtime
formattedlocation string //formattedlocation
}

```

```
isactionevent int //isactionevent
iscourseevent int //iscourseevent
iscategoryevent int //iscategoryevent
groupname string Optional //groupname
normalisedeventtype string //normalisedeventtype
normalisedeventtypetext string //normalisedeventtypetext
action object {
    name string //name
    url string //url
    itemcount int //itemcount
    actionable int //actionable
    showitemcount int //showitemcount
}
purpose string //purpose
url string //url
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [event] =>
        Array
        (
            [id] => int
            [name] => string
            [description] => string
            [descriptionformat] => int
            [location] => string
            [categoryid] => int
            [groupid] => int
            [userid] => int
            [repeatid] => int
            [eventcount] => int
            [component] => string
            [modulename] => string
            [activityname] => string
            [activitystr] => string
            [instance] => int
            [eventtype] => string
            [timestart] => int
            [timeduration] => int
            [timesort] => int
            [timeusermidnight] => int
            [visible] => int
            [timemodified] => int
            [overdue] => int
            [icon] =>
                Array
                (
                    [key] => string
                    [component] => string
                    [alttext] => string
                    [iconurl] => string
                    [iconclass] => string
                )
            [category] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [idnumber] => string
                    [description] => string
                    [parent] => int
                    [coursecount] => int
                    [visible] => int
                    [timemodified] => int
                    [depth] => int
                    [nestedname] => string
                    [url] => string
                )
            [course] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                    [idnumber] => string
                    [summary] => string
                    [summaryformat] => int
                    [startdate] => int
                    [enddate] => int
                    [visible] => int
                    [showactivitydates] => int
                    [showcompletionconditions] => int
                    [pdfexportfont] => string
                    [fullnamedisplay] => string
                    [viewurl] => string
                    [courseimage] => string
                    [progress] => int
                    [hasprogress] => int
                    [isfavourite] => int
                    [hidden] => int
                    [timeaccess] => int
                    [showshortname] => int
                    [coursecategory] => string
                )
            [subscription] =>
                Array

```



```
(  
    [displayeventsource] => int  
    [subscriptionname] => string  
    [subscriptionurl] => string  
)  
[canedit] => int  
[candelete] => int  
[deleteurl] => string  
[editurl] => string  
[viewurl] => string  
[formattedtime] => string  
[formattedlocation] => string  
[isactionevent] => int  
[iscourseevent] => int  
[iscategoryevent] => int  
[groupname] => string  
[normalisedeventtype] => string  
[normalisedeventtypeptext] => string  
[action] =>  
    Array  
        (  
            (  
                [name] => string  
                [url] => string  
                [itemcount] => int  
                [actionable] => int  
                [showitemcount] => int  
            )  
        )  
    [purpose] => string  
    [url] => string  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="event">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="location">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="categoryid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="repeatid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventcount">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="component">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="modulename">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activityname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="activitystr">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="instance">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="eventtype">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="timestart">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeduration">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timesort">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeusermidnight">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="overdue">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="icon">
          <SINGLE>
            <KEY name="key">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="component">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```
</KEY>
<KEY name="alttext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="iconclass">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="category">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="parent">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="coursecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="depth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="nestedname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="course">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summary">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="summaryformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="startdate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="enddate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showactivitydates">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showcompletionconditions">
            <VALUE>int</VALUE>
        </KEY>
```



```
</KEY>
<KEY name="pdfexportfont">
    <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
    <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="subscription">
    <SINGLE>
        <KEY name="displayeventsources">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="subscriptionname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscriptionurl">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="candelete">
    <VALUE>int</VALUE>
</KEY>
<KEY name="deleteurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="editurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedtime">
    <VALUE>string</VALUE>
</KEY>
<KEY name="formattedlocation">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isactionevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscourseevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscategoryevent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtype">
```



```

<VALUE>string</VALUE>
</KEY>
<KEY name="normalisedeventtypepetext">
    <VALUE>string</VALUE>
</KEY>
<KEY name="action">
    <SINGLE>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="actionable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showitemcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="purpose">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes



Callable from AJAX

Yes

core_change_editmode

Change the editing mode

Arguments

setmode (Required)

Set edit mode to

General structure

```
int //Set edit mode to
```

XML-RPC (PHP structure)

```
[setmode] => int
```

REST (POST parameters)

```
setmode= int
```

context (Required)

Page context id

General structure

```
int //Page context id
```

XML-RPC (PHP structure)

```
[context] => int
```

REST (POST parameters)

```
context= int
```

Response

editmode

General structure

```
//editmode
object {
success int //The edit mode was changed
}
```

XML-RPC (PHP structure)

```
Array
(
    [success] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="success">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_cohort_add_cohort_members`

Adds cohort members.

Arguments

`members` (Required)

General structure

```
list of (
object {
cohorttype object {
type string //The name of the field: id
                    (numeric value of cohortid) or idnumber (alphanumeric value of idnumber)
value string //The value of the cohort
}
usertype object {
type string //The name of the field: id
                    (numeric value of id) or username (alphanumeric value of username)
value string //The value of the cohort
}
}
```

XML-RPC (PHP structure)

```
[members] =>
    Array
    (
        [0] =>
            Array
            (
                [cohorttype] =>
                    Array
                    (
                        [type] => string
                        [value] => string
                    )
                [usertype] =>
                    Array
                    (
                        [type] => string
                        [value] => string
                    )
            )
    )
```



REST (POST parameters)

```
members[0][cohorttype][type]= string
members[0][cohorttype][value]= string
members[0][usertype][type]= string
members[0][usertype][value]= string
```

Response

General structure

```
object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
          )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_cohort_create_cohorts

Creates new cohorts.

Arguments

cohorts (Required)

General structure

```
list of (
object {
categorytype object {
type string //the name of the field: id (numeric value
                    of course category id) or idnumber (alphanumeric value of idnumber course category)
                    or system (value ignored)
value string //the value of the categorytype
}
name string //cohort name
idnumber string //cohort idnumber
description string Optional //cohort description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int Optional //cohort visible
theme string Optional //the cohort theme. The allowcohortthemes setting must be enabled on Moodle
customfields Optional //Custom fields for the cohort
list of (
object {
shortname string //The shortname of the custom field
value string //The value of the custom field
})
})
)
```

XML-RPC (PHP structure)

```
[cohorts] =>
Array
(
[0] =>
    Array
    (
        [categorytype] =>
            Array
            (
                [type] => string
                [value] => string
            )
        [name] => string
        [idnumber] => string
        [description] => string
        [descriptionformat] => int
        [visible] => int
        [theme] => string
        [customfields] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [shortname] => string
                        [value] => string
                    )
            )
    )
)
```

REST (POST parameters)

```

cohorts[0][categorytype][type]= string
cohorts[0][categorytype][value]= string
cohorts[0][name]= string
cohorts[0][idnumber]= string
cohorts[0][description]= string
cohorts[0][descriptionformat]= int
cohorts[0][visible]= int
cohorts[0][theme]= string
cohorts[0][customfields][0][shortname]= string
cohorts[0][customfields][0][value]= string

```

Response**General structure**

```

list of (
object {
id int //cohort id
name string //cohort name
idnumber string //cohort idnumber
description string //cohort description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //cohort visible
theme string Optional //cohort theme
}
)

```

XML-RPC (PHP structure)

```

Array
(
[0] =>
Array
(
[id] => int
[name] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[visible] => int
[theme] => string
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="visible">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="theme">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**core_cohort_delete_cohort_members**

Deletes cohort members.

Arguments

members (Required)

General structure
<pre>list of (object { cohortid int //cohort record id userid int //user id })</pre>

XML-RPC (PHP structure)

```
[members] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [cohortid] => int
                        [userid] => int
                    )
                )
        )
    )
```

REST (POST parameters)

```
members[0][cohortid]= int
members[0][userid]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_cohort_delete_cohorts

Deletes all specified cohorts.

**Arguments**

cohortids (Required)

General structure

```
list of (
int //cohort ID
)
```

XML-RPC (PHP structure)

```
[cohortids] =>
    Array
    (
        [
            [0] => int
        )
    )
```

REST (POST parameters)

```
cohortids[0]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_cohort_get_cohort_members

Returns cohort members.

Arguments

cohortids (Required)

General structure

```
list of (
int //Cohort ID
)
```

XML-RPC (PHP structure)

```
[cohortids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
cohortids[0]= int
```

Response

General structure

```
list of (
object {
cohortid int //cohort record id
userids list of (
int //user id
)
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[cohortid] => int
[userids] =>
Array
(
[0] => int
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="cohortid">
<VALUE>int</VALUE>
</KEY>
<KEY name="userids">
<MULTIPLE>
<VALUE>int</VALUE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_cohort_getcohorts

Returns cohort details.

Arguments

`cohortids` (Default to "Array ()")

List of cohort id. A cohort id is an integer.

General structure

```
Default to "Array"
(
)
" //List of cohort id. A cohort id is an integer.
list of (
int  //Cohort ID
)
```

XML-RPC (PHP structure)

```
[cohortids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
cohortids[0]= int
```

Response

General structure

```
list of (
object {
id int  //ID of the cohort
name string  //cohort name
idnumber string  //cohort idnumber
description string  //cohort description
descriptionformat int  //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int  //cohort visible
theme string  Optional //cohort theme
customfields  Optional //Custom fields
list of (
object {
name string  //The name of the custom field
shortname string  //The shortname of the custom field - to be able to build the field class in the code
type string  //The type of the custom field - text field, checkbox...
valueraw string  //The raw value of the custom field
value string  //The value of the custom field
}
)
)
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [name] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [visible] => int
                [theme] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [shortname] => string
                                        [type] => string
                                        [valueraw] => string
                                        [value] => string
                                    )
                        )
                )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="visible">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="theme">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="customfields">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="name">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="shortname">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="type">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="valueraw">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="value">
                            <VALUE>string</VALUE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_cohort_searchcohorts

Search for cohorts.

Arguments

query (Required)

Query string

General structure

```
string //Query string
```

XML-RPC (PHP structure)

```
[query] => string
```

REST (POST parameters)

```
query= string
```

context (Required)

General structure

```
object {
    contextid int Default to "0" //Context ID. Either use this value, or level and instanceid.
    contextlevel string Default to "" //Context Level. To be used with instanceid.
    instanceid int Default to "0" //Context instance ID. To be used with level
}
```

XML-RPC (PHP structure)

```
[context] =>
    Array
        (
            [contextid] => int
            [contextlevel] => string
            [instanceid] => int
        )
```

REST (POST parameters)

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```



includes (Default to "parents")

What other contexts to fetch the frameworks from. (all, parents, self)

General structure

```
string Default to "parents" //What other contexts to fetch the frameworks from. (all, parents, self)
```

XML-RPC (PHP structure)

```
[includes] => string
```

REST (POST parameters)

```
includes= string
```

limitfrom (Default to "0")

limitfrom we are fetching the records from

General structure

```
int Default to "0" //Limit from we are fetching the records from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "25")

Number of records to fetch

General structure

```
int Default to "25" //Number of records to fetch
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

Response**General structure**

```
object {
cohorts list of (
object {
id int //ID of the cohort
name string //cohort name
idnumber string //cohort idnumber
description string //cohort description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //cohort visible
theme string Optional //cohort theme
customfields Optional //Custom fields
list of (
object {
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
type string //The type of the custom field - text field, checkbox...
valueraw string //The raw value of the custom field
value string //The value of the custom field
}
)
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [cohorts] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [idnumber] => string
                            [description] => string
                            [descriptionformat] => int
                            [visible] => int
                            [theme] => string
                            [customfields] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [name] => string
                                                    [shortname] => string
                                                    [type] => string
                                                    [valueraw] => string
                                                    [value] => string
                                                )
                                    )
                        )
                )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="cohorts">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="theme">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="customfields">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="shortname">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="type">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="valueraw">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="value">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_cohort_update_cohorts

Updates existing cohorts.

Arguments

cohorts (Required)

General structure

```
list of (
object {
id int //ID of the cohort
categorytype object {
type string //the name of the field: id (numeric value
               of course category id) or idnumber (alphanumeric value of idnumber course category)
               or system (value ignored)
value string //the value of the categorytype
}
name string //cohort name
idnumber string //cohort idnumber
description string Optional //cohort description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int Optional //cohort visible
theme string Optional //the cohort theme. The allowcohortthemes setting must be enabled on Moodle
customfields Optional //Custom fields for the cohort
list of (
object {
shortname string //The shortname of the custom field
value string //The value of the custom field
}
)}
)}
```

XML-RPC (PHP structure)

```
[cohorts] =>
Array
(
[0] =>
Array
(
[id] => int
[categorytype] =>
Array
(
[type] => string
[value] => string
)
[name] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[visible] => int
[theme] => string
[customfields] =>
Array
(
[0] =>
Array
(
[shortname] => string
[value] => string
)
)
)
)
```

REST (POST parameters)

```

cohorts[0][id]= int
cohorts[0][categorytype][type]= string
cohorts[0][categorytype][value]= string
cohorts[0][name]= string
cohorts[0][idnumber]= string
cohorts[0][description]= string
cohorts[0][descriptionformat]= int
cohorts[0][visible]= int
cohorts[0][theme]= string
cohorts[0][customfields][0][shortname]= string
cohorts[0][customfields][0][value]= string

```

Response**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_comment_add_comments

Adds a comment or comments.

Argumentscomments (Required) ^**General structure**

```

list of (
object {
contextlevel string //contextlevel system, course, user...
instanceid int //the id of item associated with the contextlevel
component string //component
content string //component
itemid int //associated id
area string Default to "" //string comment area
}
)

```

XML-RPC (PHP structure)

```
[comments] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [contextlevel] => string
                        [instanceid] => int
                        [component] => string
                        [content] => string
                        [itemid] => int
                        [area] => string
                    )
                )
        )
    )
```

REST (POST parameters)

```
comments[0][contextlevel]= string
comments[0][instanceid]= int
comments[0][component]= string
comments[0][content]= string
comments[0][itemid]= int
comments[0][area]= string
```

Response**General structure**

```
list of (
    //comment
    object {
        id int //Comment ID
        content string //The content text formatted
        format int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
        timecreated int //Time created (timestamp)
        strftimeformat string //Time format
        profileurl string //URL profile
        fullname string //fullname
        time string //Time in human format
        avatar string //HTML user picture
        userid int //User ID
        delete int Optional //Permission to delete=true/false
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [
                        [id] => int
                        [content] => string
                        [format] => int
                        [timecreated] => int
                        [strftimeformat] => string
                        [profileurl] => string
                        [fullname] => string
                        [time] => string
                        [avatar] => string
                        [userid] => int
                        [delete] => int
                    )
                )
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="content">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="format">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="strftimeformat">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="time">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="avatar">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="userid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="delete">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

core_comment_delete_comments

Deletes a comment or comments.

Arguments

comments (Required)

General structure

```
list of (
int Default to "0" //id of the comment
)
```

XML-RPC (PHP structure)

```
[comments] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
comments[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="item">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
<VALUE>string</VALUE>
</KEY>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_comment_get_comments

Returns comments.

Arguments**contextlevel** (Required)

contextlevel system, course, user...

General structure

```
string //contextLevel system, course, user...
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Required)

the Instance id of item associated with the context level

**General structure**

```
int //the Instance id of item associated with the context Level
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

itemid (Required)

associated id

General structure

```
int //associated id
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

area (Default to "")

string comment area

General structure

```
string Default to "" //string comment area
```

XML-RPC (PHP structure)

```
[area] => string
```

REST (POST parameters)

```
area= string
```



page (Default to "0")

page number (0 based)

General structure

```
int Default to "0" //page number (0 based)
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

sortdirection (Default to "DESC")

Sort direction: ASC or DESC

General structure

```
string Default to "DESC" //Sort direction: ASC or DESC
```

XML-RPC (PHP structure)

```
[sortdirection] => string
```

REST (POST parameters)

```
sortdirection= string
```

Response**General structure**

```
object {
comments //List of comments
list of (
//comment
object {
id int //Comment ID
content string //The content text formatted
format int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
timecreated int //Time created (timestamp)
strftimeformat string //Time format
profileurl string //URI profile
fullname string //fullname
time string //Time in human format
avatar string //HTML user picture
userid int //User ID
delete int Optional //Permission to delete=true/false
}
)count int Optional //Total number of comments.
perpage int Optional //Number of comments per page.
canpost int Optional //Whether the user can post in this comment area.
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [
        [comments] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [content] => string
                                [format] => int
                                [timecreated] => int
                                [strftimeformat] => string
                                [profileurl] => string
                                [fullname] => string
                                [time] => string
                                [avatar] => string
                                [userid] => int
                                [delete] => int
                            )
                )
            )
        [count] => int
        [perpage] => int
        [canpost] => int
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
    )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="comments">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="content">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="strftimeformat">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="time">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="avatar">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="delete">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="perpage">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_competency_add_competency_to_course

Add the competency to a course

Arguments**courseid** (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

**XML-RPC (PHP structure)**

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_add_competency_to_plan

Add the competency to a learning plan

Arguments

planid (Required)

The plan id

**General structure**

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_add_competency_to_template

Add the competency to a template

Arguments

templateid (Required)

The template id

General structure

```
int //The template id
```

XML-RPC (PHP structure)

```
[templateid] => int
```

REST (POST parameters)

```
templateid= int
```

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_add_related_competency`

Adds a related competency

Arguments

`competencyid` (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

`relatedcompetencyid` (Required)

The related competency id

General structure

```
int //The related competency id
```

XML-RPC (PHP structure)

```
[relatedcompetencyid] => int
```

REST (POST parameters)

```
relatedcompetencyid= int
```



Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_approve_plan

Approve a plan.

Arguments

id (Required)

The plan ID

General structure

int //The plan ID

XML-RPC (PHP structure)

[id] => int

**REST (POST parameters)**

id= int

Response

The success

General structure

int //The success

XML-RPC (PHP structure)

int

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_competency_framework_viewed

Log event competency framework viewed

Arguments

id (Required)

The competency framework id

General structure

```
int //The competency framework id
```

XML-RPC (PHP structure)

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

Response

True if the event competency framework was logged

General structure

```
int //True if the event competency framework was Logged
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_competency_viewed

Log event competency viewed

Arguments

id (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

Response

True if the event competency viewed was logged

General structure

```
int //True if the event competency viewed was Logged
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_complete_plan

Complete learning plan.

Arguments**planid** (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

**REST (POST parameters)**

```
planid= int
```

Response

True if completing learning plan was successful

General structure

```
int //True if completing Learning plan was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_competencies

Count a list of a competencies.

Arguments

filters (Required)

General structure

```
list of (
object {
column string //Column name to filter by
value string //Value to filter by. Must be exact match
}
)
```

**XML-RPC (PHP structure)**

```
[filters] =>
    Array
        (
            [0] =>
                Array
                    (
                        [column] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
filters[0][column]= string
filters[0][value]= string
```

Response

The number of competencies found.

General structure

```
int //The number of competencies found.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_competencies_in_course

List the competencies in a course



Arguments

id (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The number of competencies in this course.

General structure

```
int //The number of competencies in this course.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_competencies_in_template

Count a list of a competencies for a given template.



Arguments

id (Required)

The template id

General structure

```
int //The template id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The number of competencies in this learning plan template.

General structure

```
int //The number of competencies in this Learning plan template.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_competency_frameworks

Count a list of competency frameworks.



Arguments

context (Required)

General structure

```
object {
    contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
    contextlevel string Default to "" //Context Level. To be used with instanceid.
    instanceid int Default to "0" //Context instance ID. To be used with level
}
```

XML-RPC (PHP structure)

```
[context] =>
    Array
    (
        [contextid] => int
        [contextlevel] => string
        [instanceid] => int
    )
```

REST (POST parameters)

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```

includes (Default to "children")

What other contexts to fetch the frameworks from. (children, parents, self)

General structure

```
string Default to "children" //What other contexts to fetch the frameworks from. (children, parents, self)
```

XML-RPC (PHP structure)

```
[includes] => string
```

REST (POST parameters)

```
includes= string
```

Response

The number of competency frameworks found.

General structure

```
int //The number of competency frameworks found.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_course_module_competencies

Count the competencies in a course module

Arguments**cmid** (Required)

The course module id

General structure

```
int //The course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

Response

The number of competencies found.

General structure

```
int //The number of competencies found.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_courses_using_competency

List the courses using a competency

Arguments

id (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The number of courses using this competency

General structure

```
int //The number of courses using this competency
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_templates

Count a list of a learning plan templates.

Arguments

context (Required)

General structure

```
object {
    contextid int Default to "0" //Context ID. Either use this value, or level and instanceid.
    contextlevel string Default to "" //Context Level. To be used with instanceid.
    instanceid int Default to "0" //Context instance ID. To be used with level
}
```

XML-RPC (PHP structure)

```
[context] =>
    Array
        (
            [contextid] => int
            [contextlevel] => string
            [instanceid] => int
        )
```

REST (POST parameters)

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```

includes (Default to "children")

What other contexts to fetch the frameworks from. (children, parents, self)

General structure

```
string Default to "children" //What other contexts to fetch the frameworks from. (children, parents, self)
```

XML-RPC (PHP structure)

```
[includes] => string
```

**REST (POST parameters)**

```
includes= string
```

Response

The number of learning plan templates found.

General structure

```
int //The number of Learning plan templates found.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_count_templates_using_competency

Count a list of learning plan templates for a given competency.

Arguments

id (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

Response

The number of learning plan templates using this competency

General structure

```
int //The number of Learning plan templates using this competency
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_create_competency

Creates new competencies.

Arguments

competency (Required)

General structure

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string Default to "" //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    sortorder int Default to "0" //sortorder
    parentid int Default to "0" //parentid
    path string Default to "/0/" //path
    ruleoutcome int Default to "0" //ruleoutcome
    ruletype string Default to "null" //ruletype
    ruleconfig string Default to "null" //ruleconfig
    scaleid int Default to "null" //scaleid
    scaleconfiguration string Default to "null" //scaleconfiguration
    competencyframeworkid int Default to "0" //competencyframeworkid
    timecreated int Default to "0" //timecreated
    timemodified int Default to "0" //timemodified
    usermodified int Default to "0" //usermodified
}
```

XML-RPC (PHP structure)

```
[competency] =>
    Array
        (
            [shortname] => string
            [idnumber] => string
            [description] => string
            [descriptionformat] => int
            [sortorder] => int
            [parentid] => int
            [path] => string
            [ruleoutcome] => int
            [ruletype] => string
            [ruleconfig] => string
            [scaleid] => int
            [scaleconfiguration] => string
            [competencyframeworkid] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )
```

REST (POST parameters)

```
competency[shortname]= string
competency[idnumber]= string
competency[description]= string
competency[descriptionformat]= int
competency[sortorder]= int
competency[parentid]= int
competency[path]= string
competency[ruleoutcome]= int
competency[ruletype]= string
competency[ruleconfig]= string
competency[scaleid]= int
competency[scaleconfiguration]= string
competency[competencyframeworkid]= int
competency[timecreated]= int
competency[timemodified]= int
competency[usermodified]= int
```

Response**General structure**

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN) ^
    sortorder int //sortorder
    parentid int //parentid
    path string //path
    ruleoutcome int //ruleoutcome
    ruletype string //ruletype
    ruleconfig string //ruleconfig
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    competencyframeworkid int //competencyframeworkid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
}
```

XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [idnumber] => string
    [description] => string
    [descriptionformat] => int
    [sortorder] => int
    [parentid] => int
    [path] => string
    [ruleoutcome] => int
    [ruletype] => string
    [ruleconfig] => string
    [scaleid] => int
    [scaleconfiguration] => string
    [competencyframeworkid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="sortorder">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="parentid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="path">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="ruleoutcome">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ruletype">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="ruleconfig">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="scaleid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleconfiguration">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="competencyframeworkid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_create_competency_framework

Creates new competency frameworks.

Arguments

competencyframework (Required)

```
General structure

object {
    shortname string //shortname
    idnumber string //idnumber
    description string Default to "" //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    visible int Default to "1" //visible
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    contextid int Optional //The context id
    contextlevel string Optional //The context level
    instanceid int Optional //The Instance id
    taxonomies string Default to "" //taxonomies
    timecreated int Default to "0" //timecreated
    timemodified int Default to "0" //timemodified
    usermodified int Default to "0" //usermodified
}
```

XML-RPC (PHP structure)

```
[competencyframework] =>
    Array
        (
            [shortname] => string
            [idnumber] => string
            [description] => string
            [descriptionformat] => int
            [visible] => int
            [scaleid] => int
            [scaleconfiguration] => string
            [contextid] => int
            [contextlevel] => string
            [instanceid] => int
            [taxonomies] => string
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )
```

**REST (POST parameters)**

```
competencyframework[shortname]= string
competencyframework[idnumber]= string
competencyframework[description]= string
competencyframework[descriptionformat]= int
competencyframework[visible]= int
competencyframework[scaleid]= int
competencyframework[scaleconfiguration]= string
competencyframework[contextid]= int
competencyframework[contextlevel]= string
competencyframework[instanceid]= int
competencyframework[taxonomies]= string
competencyframework[timecreated]= int
competencyframework[timemodified]= int
competencyframework[usermodified]= int
```

Response

General structure

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    visible int //visible
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    contextid int //contextid
    taxonomies string //taxonomies
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    canmanage int //canmanage
    competenciescount int //competenciescount
    contextname string //contextname
    contextnamenoprefix string //contextnamenoprefix
}
```

XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [idnumber] => string
    [description] => string
    [descriptionformat] => int
    [visible] => int
    [scaleid] => int
    [scaleconfiguration] => string
    [contextid] => int
    [taxonomies] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [canmanage] => int
    [competenciescount] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleconfiguration">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="taxonomies">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="competenciescount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_create_plan

Creates a learning plan.

Arguments

plan (Required)

General structure

```
object {
    name string //name
    description string Default to "" //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    userid int //userid
    templateid int Default to "null" //templateid
    origtemplateid int Default to "null" //origtemplateid
    status int Default to "0" //status
    duedate int Default to "0" //duedate
    reviewerid int Default to "null" //reviewerid
    timecreated int Default to "0" //timecreated
    timemodified int Default to "0" //timemodified
    usermodified int Default to "0" //usermodified
}
```

XML-RPC (PHP structure)

```
[plan] =>
    Array
        (
            [name] => string
            [description] => string
            [descriptionformat] => int
            [userid] => int
            [templateid] => int
            [origtemplateid] => int
            [status] => int
            [duedate] => int
            [reviewerid] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )
```

REST (POST parameters)

```
plan[name]= string
plan[description]= string
plan[descriptionformat]= int
plan[userid]= int
plan[templateid]= int
plan[origtemplateid]= int
plan[status]= int
plan[duedate]= int
plan[reviewerid]= int
plan[timecreated]= int
plan[timemodified]= int
plan[usermodified]= int
```

Response

General structure

```

object {
    name string //name
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    userid int //userid
    templateid int //templateid
    origtemplateid int //origtemplateid
    status int //status
    duedate int //duedate
    reviewerid int //reviewerid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    statusname string //statusname
    isbasedontemplate int //isbasedontemplate
    canmanage int //canmanage
    canrequestreview int //canrequestreview
    canreview int //canreview
    canbeedited int //canbeedited
    isactive int //isactive
    isdraft int //isdraft
    iscompleted int //iscompleted
    isinreview int //isinreview
    iswaitingforreview int //iswaitingforreview
    isreopenallowed int //isreopenallowed
    incompleteallowed int //incompleteallowed
    isunlinkallowed int //isunlinkallowed
    isrequestreviewallowed int //isrequestreviewallowed
    iscancelreviewrequestallowed int //iscancelreviewrequestallowed
    isstartreviewallowed int //isstartreviewallowed
    isstopreviewallowed int //isstopreviewallowed
    isapprovewallowed int //isapprovewallowed
    isunapprovewallowed int //isunapprovewallowed
    duedateformatted string //duedateformatted
    commentarea object {
        component string //component
        commentarea string //commentarea
    }
    itemid int //itemid
    courseid int //courseid
    contextid int //contextid
    cid string //cid
    autostart int //autostart
    canpost int //canpost
    canview int //canview
    count int //count
    collapsediconkey string //collapsediconkey
    displaytotalcount int //displaytotalcount
    displaycancel int //displaycancel
    fullwidth int //fullwidth
    linktext string //linktext
    notoggle int //notoggle
    template string //template
    canpostorhascomments int //canpostorhascomments
}
reviewer object {
    id int //id
    email string //email
    idnumber string //idnumber
    phone1 string //phone1
    phone2 string //phone2
    department string //department
    institution string //institution
    fullname string //fullname
    identity string //identity
    profileurl string //profileurl
    profileimageurl string //profileimageurl
    profileimageurlsmall string //profileimageurlsmall
}
template object {
    shortname string //shortname
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    duedate int //duedate
    visible int //visible
    contextid int //contextid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    duedateformatted string //duedateformatted
}

```



```
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [name] => string
    [description] => string
    [descriptionformat] => int
    [userid] => int
    [templateid] => int
    [origtemplateid] => int
    [status] => int
    [duedate] => int
    [reviewerid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [statusname] => string
    [isbasedontemplate] => int
    [canmanage] => int
    [canrequestreview] => int
    [canreview] => int
    [canbeedited] => int
    [isactive] => int
    [isdraft] => int
    [iscompleted] => int
    [isinreview] => int
    [iswaitingforreview] => int
    [isreopenallowed] => int
    [iscompleteallowed] => int
    [isunlinkallowed] => int
    [isrequestreviewallowed] => int
    [iscancelreviewrequestallowed] => int
    [isstartreviewallowed] => int
    [isstopreviewallowed] => int
    [isapproveallowed] => int
    [isunapproveallowed] => int
    [duedateformatted] => string
    [commentarea] =>
        Array
        (
            [component] => string
            [commentarea] => string
            [itemid] => int
            [courseid] => int
            [contextid] => int
            [cid] => string
            [autostart] => int
            [canpost] => int
            [canview] => int
            [count] => int
            [collapsediconkey] => string
            [displaytotalcount] => int
            [displaycancel] => int
            [fullwidth] => int
            [linktext] => string
            [notoggle] => int
            [template] => string
            [canpostorhascomments] => int
        )
    [reviewer] =>
        Array
        (
            [id] => int
            [email] => string
            [idnumber] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [fullname] => string
            [identity] => string
            [profileurl] => string
            [profileimageurl] => string
            [profileimageurlsmall] => string
        )
    [template] =>
        Array
        (
            [shortname] => string
            [description] => string
            [descriptionformat] => int
        )
)

```



```
[duedate] => int
[visible] => int
[contextid] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="templateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="origtemplateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="reviewerid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="isbasedontemplate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canrequestreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canbeedited">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isactive">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isdraft">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isinreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iswaitingforreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isreopenallowed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleteallowed">
      <VALUE>int</VALUE>
    </KEY>
```



```
<KEY name="isunlinkallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="commentarea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="autostart">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canview">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="collapsediconkey">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="displaytotalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="displaycancel">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullwidth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="linktext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="notoggle">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="template">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="canpostorhascomments">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="reviewer">
```



```
<SINGLE>
    <KEY name="id">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="email">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone1">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone2">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="department">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="institution">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="fullname">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="identity">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</KEY>
<KEY name="template">
    <SINGLE>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedateformatted">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="cohortscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="planscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanage">
            <VALUE>int</VALUE>
        </KEY>
```



```

<KEY name="canread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_create_template

Creates new learning plan templates.

Arguments

template (Required)



General structure

```

object {
    shortname string //shortname
    description string Default to "" //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    duedate int Default to "0" //duedate
    visible int Default to "1" //visible
    contextid int Optional //The context id
    contextlevel string Optional //The context Level
    instanceid int Optional //The Instance id
    timemodified int Default to "0" //timemodified
    usermodified int Default to "0" //usermodified
}

```

XML-RPC (PHP structure)

```
[template] =>
  Array
  (
    [shortname] => string
    [description] => string
    [descriptionformat] => int
    [duedate] => int
    [visible] => int
    [contextid] => int
    [contextlevel] => string
    [instanceid] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
  )
```

REST (POST parameters)

```
template[shortname]= string
template[description]= string
template[descriptionformat]= int
template[duedate]= int
template[visible]= int
template[contextid]= int
template[contextlevel]= string
template[instanceid]= int
template[timecreated]= int
template[timemodified]= int
template[usermodified]= int
```

Response**General structure**

```
object {
  shortname string //shortname
  description string //description
  descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
  duedate int //duedate
  visible int //visible
  contextid int //contextid
  id int //id
  timecreated int //timecreated
  timemodified int //timemodified
  usermodified int //usermodified
  duedateformatted string //duedateformatted
  cohortscount int //cohortscount
  planscount int //planscount
  canmanage int //canmanage
  canread int //canread
  contextname string //contextname
  contextnamenoprefix string //contextnamenoprefix
}
```



XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [description] => string
    [descriptionformat] => int
    [duedate] => int
    [visible] => int
    [contextid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [duedateformatted] => string
    [cohortscount] => int
    [planscount] => int
    [canmanage] => int
    [canread] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedateformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="cohortscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="planscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_create_user_evidence_competency

Create an evidence of prior learning relationship with a competency.

Arguments**userevidenceid** (Required)

The user evidence ID.

General structure

```
int //The user evidence ID.
```

XML-RPC (PHP structure)

```
[userevidenceid] => int
```

REST (POST parameters)

```
userevidenceid= int
```

competencyid (Required)

The competency ID.

General structure

```
int //The competency ID.
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

**Response****General structure**

```
object {
    userevidenceid int //userevidenceid
    competencyid int //competencyid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
}
```

XML-RPC (PHP structure)

```
Array
(
    [userevidenceid] => int
    [competencyid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="userevidenceid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="competencyid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_delete_competency

Delete a competency.

Arguments

id (Required)

Data base record id for the competency

General structure

```
int //Data base record id for the competency
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_delete_competency_framework

Delete a competency framework.

Arguments

id (Required)

Data base record id for the framework

General structure

```
int //Data base record id for the framework
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_delete_evidence

Delete an evidence

Arguments

id (Required)

The evidence ID

General structure

```
int //The evidence ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_delete_plan

Delete a learning plan.

Arguments

id (Required)

Data base record id for the learning plan

General structure

```
int //Data base record id for the Learning plan
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_delete_template

Delete a learning plan template.

Arguments

id (Required)

Data base record id for the template

General structure

```
int //Data base record id for the template
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

deleteplans (Required)

Boolean to indicate if plans must be deleted

General structure

```
int //Boolean to indicate if plans must be deleted
```

XML-RPC (PHP structure)

```
[deleteplans] => int
```

REST (POST parameters)

```
deleteplans= int
```

Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_delete_user_evidence`

Delete an evidence of prior learning.

Arguments

`id` (Required)

The user evidence ID.

General structure

```
int //The user evidence ID.
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_delete_user_evidence_competency`

Delete an evidence of prior learning relationship with a competency.

Arguments

`userevidenceid` (Required)

The user evidence ID.

General structure

```
int //The user evidence ID.
```

XML-RPC (PHP structure)

```
[userevidenceid] => int
```

REST (POST parameters)

```
userevidenceid= int
```

`competencyid` (Required)

The competency ID.

General structure

```
int //The competency ID.
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```



Response

True if the delete was successful

General structure

```
int //True if the delete was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_duplicate_competency_framework

Duplicate a competency framework.

Arguments

id (Required)

Data base record id for the framework

General structure

```
int //Data base record id for the framework
```

XML-RPC (PHP structure)

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

Response

General structure

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    visible int //visible
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    contextid int //contextid
    taxonomies string //taxonomies
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    canmanage int //canmanage
    competenciescount int //competenciescount
    contextname string //contextname
    contextnamenoprefix string //contextnamenoprefix
}
```

XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [idnumber] => string
    [description] => string
    [descriptionformat] => int
    [visible] => int
    [scaleid] => int
    [scaleconfiguration] => string
    [contextid] => int
    [taxonomies] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [canmanage] => int
    [competenciescount] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleconfiguration">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="taxonomies">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="competenciescount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_duplicate_template

Duplicate learning plan template.

Arguments

id (Required)

The template id

General structure

```
int //The template id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response**General structure**

```
object {
    shortname string //shortname
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    duedate int //duedate
    visible int //visible
    contextid int //contextid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    duedateformatted string //duedateformatted
    cohortscount int //cohortscount
    planscount int //planscount
    canmanage int //canmanage
    canread int //canread
    contextname string //contextname
    contextnamenoprefix string //contextnamenoprefix
}
```



XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [description] => string
    [descriptionformat] => int
    [duedate] => int
    [visible] => int
    [contextid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [duedateformatted] => string
    [cohortscount] => int
    [planscount] => int
    [canmanage] => int
    [canread] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedateformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="cohortscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="planscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_get_scale_values

Fetch the values for a specific scale

Arguments**scaleid** (Required)

The scale id

General structure

int //The scale id

XML-RPC (PHP structure)

[scaleid] => int

REST (POST parameters)

scaleid= int

Response**General structure**

```
list of (
object {
id int //Scale value ID
name string //Scale value name
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[id] => int
[name] => string
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_grade_competency`

Grade a competency.

Arguments**userid** (Required)

User ID

General structure

int //User ID

XML-RPC (PHP structure)

[userid] => int

REST (POST parameters)

userid= int

competencyid (Required)

Competency ID

**General structure**

int //Competency ID

XML-RPC (PHP structure)

[competencyid] => int

REST (POST parameters)

competencyid= int

grade (Required)

New grade

General structure

```
int //New grade
```

XML-RPC (PHP structure)

```
[grade] => int
```

REST (POST parameters)

```
grade= int
```

note (Default to "null")

A note to attach to the evidence

General structure

```
string Default to "null" //A note to attach to the evidence
```

XML-RPC (PHP structure)

```
[note] => string
```

REST (POST parameters)

```
note= string
```

Response**General structure**

```
object {
    usercompetencyid int //usercompetencyid
    contextid int //contextid
    action int //action
    actionuserid int //actionuserid
    descidentifier string //descidentifier
    desccomponent string //desccomponent
    desca string //desca
    url string //url
    grade int //grade
    note string //note
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    actionuser object {
        id int //id
        email string //email
        idnumber string //idnumber
        phone1 string //phone1
        phone2 string //phone2
        department string //department
        institution string //institution
        fullname string //fullname
        identity string //identity
        profileurl string //profileurl
        profileimageurl string //profileimageurl
        profileimageurlsmall string //profileimageurlsmall
    }
    description string //description
    gradename string //gradename
    userdate string //userdate
    candelete int //candelete
}
```



XML-RPC (PHP structure)

```
Array
(
    [usercompetencyid] => int
    [contextid] => int
    [action] => int
    [actionuserid] => int
    [descidentifier] => string
    [desccomponent] => string
    [desca] => string
    [url] => string
    [grade] => int
    [note] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [actionuser] =>
        Array
            (
                [id] => int
                [email] => string
                [idnumber] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [fullname] => string
                [identity] => string
                [profileurl] => string
                [profileimageurl] => string
                [profileimageurlsmall] => string
            )
        [description] => string
        [gradename] => string
        [userdate] => string
        [candelete] => int
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="usercompetencyid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="action">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuserid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="descidentifier">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desccomponent">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desca">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="grade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="note">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuser">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="email">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="identity">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
```



```

<VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
<VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
<VALUE>string</VALUE>
</KEY>
<KEY name="candele"t>
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_grade_competency_in_course

Grade a competency from the course page.

Arguments

courseid (Required)

Course id



General structure

```
int //Course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Required)

User id

General structure

```
int //User id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

competencyid (Required)

Competency id

General structure

```
int //Competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

grade (Required)

New grade

General structure

```
int //New grade
```

XML-RPC (PHP structure)

```
[grade] => int
```

REST (POST parameters)

```
grade= int
```



note (Default to "null")

A note to attach to the evidence

General structure

```
string Default to "null" //A note to attach to the evidence
```

XML-RPC (PHP structure)

```
[note] => string
```

REST (POST parameters)

```
note= string
```

Response

```
General structure

object {
    usercompetencyid int //usercompetencyid
    contextid int //contextid
    action int //action
    actionuserid int //actionuserid
    descidentifier string //descidentifier
    desccomponent string //desccomponent
    desca string //desca
    url string //url
    grade int //grade
    note string //note
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    actionuser object {
        id int //id
        email string //email
        idnumber string //idnumber
        phone1 string //phone1
        phone2 string //phone2
        department string //department
        institution string //institution
        fullname string //fullname
        identity string //identity
        profileurl string //profileurl
        profileimageurl string //profileimageurl
        profileimageurlsmall string //profileimageurlsmall
    }
    description string //description
    gradename string //gradename
    userdate string //userdate
    candelete int //candelete
}
```

XML-RPC (PHP structure)

```
Array
(
    [usercompetencyid] => int
    [contextid] => int
    [action] => int
    [actionuserid] => int
    [descidentifier] => string
    [desccomponent] => string
    [desca] => string
    [url] => string
    [grade] => int
    [note] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [actionuser] =>
        Array
        (
            [id] => int
            [email] => string
            [idnumber] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [fullname] => string
            [identity] => string
            [profileurl] => string
            [profileimageurl] => string
            [profileimageurlsmall] => string
        )
    [description] => string
    [gradename] => string
    [userdate] => string
    [candelete] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="usercompetencyid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="action">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuserid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="descidentifier">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desccomponent">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desca">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="grade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="note">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuser">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="email">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="identity">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
```



```

<VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
<VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
<VALUE>string</VALUE>
</KEY>
<KEY name="candelelete">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_grade_competency_in_plan

Grade a competency from the user plan page.

Arguments

planid (Required)

Plan id



General structure

```
int //Plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

competencyid (Required)

Competency id

General structure

```
int //Competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

grade (Required)

New grade

General structure

```
int //New grade
```

XML-RPC (PHP structure)

```
[grade] => int
```

REST (POST parameters)

```
grade= int
```

note (Default to "null")

A note to attach to the evidence

General structure

```
string Default to "null" //A note to attach to the evidence
```

XML-RPC (PHP structure)

```
[note] => string
```

REST (POST parameters)

```
note= string
```

**Response**

General structure

```

object {
usercompetencyid int //usercompetencyid
contextid int //contextid
action int //action
actionuserid int //actionuserid
descidentifier string //descidentifier
desccomponent string //desccomponent
desca string //desca
url string //url
grade int //grade
note string //note
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
actionuser object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
description string //description
gradename string //gradename
userdate string //userdate
candelete int //candelete
}

```

XML-RPC (PHP structure)

```

Array
(
    [usercompetencyid] => int
    [contextid] => int
    [action] => int
    [actionuserid] => int
    [descidentifier] => string
    [desccomponent] => string
    [desca] => string
    [url] => string
    [grade] => int
    [note] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [actionuser] =>
        Array
        (
            [id] => int
            [email] => string
            [idnumber] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [fullname] => string
            [identity] => string
            [profileurl] => string
            [profileimageurl] => string
            [profileimageurlsmall] => string
        )
    [description] => string
    [gradename] => string
    [userdate] => string
    [candelete] => int
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="usercompetencyid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="action">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuserid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="descidentifier">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desccomponent">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="desca">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="grade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="note">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="actionuser">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="email">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="identity">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
```



```

<VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
<VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
<VALUE>string</VALUE>
</KEY>
<KEY name="candelelete">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_competencies

Load a list of competencies.

Arguments

filters (Required)



General structure

```

list of (
object {
column string //Column name to filter by
value string //Value to filter by. Must be exact match
}
)

```

XML-RPC (PHP structure)

```

[filters] =>
Array
(
[0] =>
Array
(
[column] => string
[value] => string
)
)

```

REST (POST parameters)

```
filters[0][column]= string
filters[0][value]= string
```

sort (Default to "")

Column to sort by.

General structure

```
string Default to "" //Column to sort by.
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

order (Default to "")

Sort direction. Should be either ASC or DESC

General structure

```
string Default to "" //Sort direction. Should be either ASC or DESC
```

XML-RPC (PHP structure)

```
[order] => string
```

REST (POST parameters)

```
order= string
```

skip (Default to "0")

Skip this number of records before returning results

General structure

```
int Default to "0" //Skip this number of records before returning results
```

**XML-RPC (PHP structure)**

```
[skip] => int
```

REST (POST parameters)

```
skip= int
```

limit (Default to "0")

Return this number of records at most.

General structure

```
int Default to "0" //Return this number of records at most.
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

limit= int

Response

General structure

```

list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
)

```

XML-RPC (PHP structure)

```

Array
(
[0] =>
    Array
        (
            [shortname] => string
            [idnumber] => string
            [description] => string
            [descriptionformat] => int
            [sortorder] => int
            [parentid] => int
            [path] => string
            [ruleoutcome] => int
            [ruletype] => string
            [ruleconfig] => string
            [scaleid] => int
            [scaleconfiguration] => string
            [competencyframeworkid] => int
            [id] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="sortorder">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="parentid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="path">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleoutcome">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="ruletype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleconfig">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="scaleid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scaleconfiguration">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="competencyframeworkid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_competencies_in_template

Load a list of competencies for a given template.

Arguments

id (Required)

The template id

General structure

```
int //The template id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
)
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [shortname] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [sortorder] => int
                [parentid] => int
                [path] => string
                [ruleoutcome] => int
                [ruletype] => string
                [ruleconfig] => string
                [scaleid] => int
                [scaleconfiguration] => string
                [competencyframeworkid] => int
                [id] => int
                [timecreated] => int
                [timemodified] => int
                [usermodified] => int
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="sortorder">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="parentid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="path">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleoutcome">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="ruletype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleconfig">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="scaleid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scaleconfiguration">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="competencyframeworkid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_competency_frameworks

Load a list of competency frameworks.

Arguments**sort** (Default to "shortname")

Column to sort by.

General structure

```
string Default to "shortname" //Column to sort by.
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

order (Default to "")

Sort direction. Should be either ASC or DESC

General structure

```
string Default to "" //Sort direction. Should be either ASC or DESC
```

XML-RPC (PHP structure)

```
[order] => string
```

REST (POST parameters)

```
order= string
```

**skip** (Default to "0")

Skip this number of records before returning results

General structure

```
int Default to "0" //Skip this number of records before returning results
```

XML-RPC (PHP structure)

```
[skip] => int
```

REST (POST parameters)

```
skip= int
```

limit (Default to "0")

Return this number of records at most.

General structure

```
int Default to "0" //Return this number of records at most.
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

context (Required)

General structure

```
object {
    contextid int Default to "0" //Context ID. Either use this value, or level and instanceid.
    contextlevel string Default to "" //Context Level. To be used with instanceid.
    instanceid int Default to "0" //Context instance ID. To be used with level
}
```

XML-RPC (PHP structure)

```
[context] =>
    Array
    (
        [contextid] => int
        [contextlevel] => string
        [instanceid] => int
    )
```

REST (POST parameters)

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```

includes (Default to "children")

What other contexts to fetch the frameworks from. (children, parents, self)

General structure

```
string Default to "children" //What other contexts to fetch the frameworks from. (children, parents, self)
```

**XML-RPC (PHP structure)**

```
[includes] => string
```

REST (POST parameters)

```
includes= string
```

onlyvisible (Default to "")

Only visible frameworks will be returned if visible true

General structure

```
int Default to "" //Only visible frameworks will be returned if visible true
```

XML-RPC (PHP structure)

```
[onlyvisible] => int
```

REST (POST parameters)

```
onlyvisible= int
```

query (Default to "")

A query string to filter the results

General structure

```
string Default to "" //A query string to filter the results
```

XML-RPC (PHP structure)

```
[query] => string
```

REST (POST parameters)

```
query= string
```

Response**General structure**

```
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //visible
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
contextid int //contextid
taxonomies string //taxonomies
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
)
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [shortname] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [visible] => int
                [scaleid] => int
                [scaleconfiguration] => string
                [contextid] => int
                [taxonomies] => string
                [id] => int
                [timecreated] => int
                [timemodified] => int
                [usermodified] => int
                [canmanage] => int
                [competenciescount] => int
                [contextname] => string
                [contextnamenoprefix] => string
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scaleid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scaleconfiguration">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="taxonomies">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmanage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="competenciescount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contextnamenoprefix">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_course_competencies

List the competencies in a course

Arguments

id (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response**General structure**

```
list of (
object {
competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
coursecompetency object {
courseid int //courseid
competencyid int //competencyid
sortorder int //sortorder
ruleoutcome int //ruleoutcome
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [competency] =>
                    Array
                        (
                            [shortname] => string
                            [idnumber] => string
                            [description] => string
                            [descriptionformat] => int
                            [sortorder] => int
                            [parentid] => int
                            [path] => string
                            [ruleoutcome] => int
                            [ruletype] => string
                            [ruleconfig] => string
                            [scaleid] => int
                            [scaleconfiguration] => string
                            [competencyframeworkid] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
                [coursecompetency] =>
                    Array
                        (
                            [courseid] => int
                            [competencyid] => int
                            [sortorder] => int
                            [ruleoutcome] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="competency">
        <SINGLE>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="description">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="descriptionformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="parentid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="path">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleoutcome">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="ruletype">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleconfig">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="scaleid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="scaleconfiguration">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="competencyframeworkid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usermodified">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </KEY>
    <KEY name="coursecompetency">
      <SINGLE>
        <KEY name="courseid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="competencyid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortorder">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="ruleoutcome">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
  </MULTIPLE>
</RESPONSE>
```



```

        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_course_module_competencies

List the competencies in a course module

Arguments

cmid (Required)

The course module id

General structure

int //The course module id

XML-RPC (PHP structure)

[cmid] => int

REST (POST parameters)

cmid= int

Response

General structure

```
list of (
object {
competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
coursemodulecompetency object {
cmid int //cmid
competencyid int //competencyid
sortorder int //sortorder
ruleoutcome int //ruleoutcome
overridegrade int //overridegrade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
}
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [competency] =>
                    Array
                        (
                            [shortname] => string
                            [idnumber] => string
                            [description] => string
                            [descriptionformat] => int
                            [sortorder] => int
                            [parentid] => int
                            [path] => string
                            [ruleoutcome] => int
                            [ruletype] => string
                            [ruleconfig] => string
                            [scaleid] => int
                            [scaleconfiguration] => string
                            [competencyframeworkid] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
                [coursemodulecompetency] =>
                    Array
                        (
                            [cmid] => int
                            [competencyid] => int
                            [sortorder] => int
                            [ruleoutcome] => int
                            [overridegrade] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="competency">
        <SINGLE>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="description">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="descriptionformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="parentid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="path">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleoutcome">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="ruletype">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleconfig">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="scaleid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="scaleconfiguration">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="competencyframeworkid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usermodified">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </KEY>
    <KEY name="coursemodulecompetency">
      <SINGLE>
        <KEY name="cmid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="competencyid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortorder">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="ruleoutcome">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="overridegrade">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
  </MULTIPLE>
</RESPONSE>
```



```

</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_plan_competencies

List the competencies in a plan

Arguments

id (Required)

The plan ID.

General structure

```
int //The plan ID.
```



XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```

list of (
object {
competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
usercompetency object {
userid int //userid
competencyid int //competencyid
status int //status
reviewerid int //reviewerid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canrequestreview int //canrequestreview
canreview int //canreview
gradename string //gradename
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
issstopreviewallowed int //issstopreviewallowed
isstatusidle int //isstatusidle
isstatusinreview int //isstatusinreview
issstatuswaitingforreview int //issstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
usercompetencyplan object {
userid int //userid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
planid int //planid
sortorder int //sortorder
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
}
}
)

```



XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [competency] =>
                    Array
                        (
                            [shortname] => string
                            [idnumber] => string
                            [description] => string
                            [descriptionformat] => int
                            [sortorder] => int
                            [parentid] => int
                            [path] => string
                            [ruleoutcome] => int
                            [ruletype] => string
                            [ruleconfig] => string
                            [scaleid] => int
                            [scaleconfiguration] => string
                            [competencyframeworkid] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
                [usercompetency] =>
                    Array
                        (
                            [userid] => int
                            [competencyid] => int
                            [status] => int
                            [reviewerid] => int
                            [proficiency] => int
                            [grade] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                            [canrequestreview] => int
                            [canreview] => int
                            [gradename] => string
                            [isrequestreviewallowed] => int
                            [iscancelreviewrequestallowed] => int
                            [isstartreviewallowed] => int
                            [isstopreviewallowed] => int
                            [isstatusidle] => int
                            [isstatusinreview] => int
                            [isstatuswaitingforreview] => int
                            [proficiencyname] => string
                            [reviewer] =>
                                Array
                                    (
                                        [id] => int
                                        [email] => string
                                        [idnumber] => string
                                        [phone1] => string
                                        [phone2] => string
                                        [department] => string
                                        [institution] => string
                                        [fullname] => string
                                        [identity] => string
                                        [profileurl] => string
                                        [profileimageurl] => string
                                        [profileimageurlsmall] => string
                                    )
                            [statusname] => string
                            [url] => string
                        )
                [usercompetencyplan] =>
                    Array
                        (
                            [userid] => int
                            [competencyid] => int
                            [proficiency] => int
                            [grade] => int
                            [planid] => int
                            [sortorder] => int
                            [id] => int
                            [timecreated] => int
                        )
            )
)

```



```
[timemodified] => int  
[usermodified] => int  
[gradename] => string  
[proficiencyname] => string  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="competency">
        <SINGLE>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="description">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="descriptionformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="parentid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="path">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleoutcome">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="ruletype">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleconfig">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="scaleid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="scaleconfiguration">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="competencyframeworkid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usermodified">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </KEY>
    <KEY name="usercompetency">
      <SINGLE>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="competencyid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="status">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="reviewerid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="proficiency">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
  </MULTIPLE>
</RESPONSE>
```



```
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canrequestreview">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="issstartreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="issstopreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
    <VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="email">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="identity">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```

<KEY name="statusname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencyplan">
    <SINGLE>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="competencyid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="proficiency">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="planid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortorder">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="gradename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="proficiencyname">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

[core_competency_list_templates](#)

Load a list of a learning plan templates.

Arguments

sort (Default to "")

Column to sort by.

General structure

```
string Default to "" //Column to sort by.
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

order (Default to "")

Sort direction. Should be either ASC or DESC

General structure

```
string Default to "" //Sort direction. Should be either ASC or DESC
```

XML-RPC (PHP structure)

```
[order] => string
```

REST (POST parameters)

```
order= string
```

skip (Default to "0")

Skip this number of records before returning results

General structure

```
int Default to "0" //Skip this number of records before returning results
```



XML-RPC (PHP structure)

```
[skip] => int
```

REST (POST parameters)

```
skip= int
```

limit (Default to "0")

Return this number of records at most.

General structure

```
int Default to "0" //Return this number of records at most.
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

context (Required)

General structure

```
object {
    contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
    contextlevel string Default to "" //Context Level. To be used with instanceid.
    instanceid int Default to "0" //Context instance ID. To be used with level
}
```

XML-RPC (PHP structure)

```
[context] =>
    Array
        (
            [contextid] => int
            [contextlevel] => string
            [instanceid] => int
        )
```

REST (POST parameters)

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```

includes (Default to "children")

What other contexts to fetch the templates from. (children, parents, self)

General structure

```
string Default to "children" //What other contexts to fetch the templates from. (children, parents, self)
```

XML-RPC (PHP structure)

```
[includes] => string
```

REST (POST parameters)

```
includes= string
```



onlyvisible (Default to "")

If should list only visible templates

General structure

```
int Default to "" //If should List only visible templates
```

XML-RPC (PHP structure)

```
[onlyvisible] => int
```

REST (POST parameters)

```
onlyvisible= int
```

Response

```
General structure

list of (
object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
        (
            [shortname] => string
            [description] => string
            [descriptionformat] => int
            [duedate] => int
            [visible] => int
            [contextid] => int
            [id] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
            [duedateformatted] => string
            [cohortscount] => int
            [planscount] => int
            [canmanage] => int
            [canread] => int
            [contextname] => string
            [contextnamenoprefix] => string
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="duedate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="duedateformatted">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="cohortscount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="planscount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmanage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canread">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contextname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="contextnamenoprefix">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_templates_using_competency

Load a list of a learning plan templates for a given competency.

Arguments

id (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```
list of (
object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
)
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [shortname] => string
                [description] => string
                [descriptionformat] => int
                [duedate] => int
                [visible] => int
                [contextid] => int
                [id] => int
                [timecreated] => int
                [timemodified] => int
                [usermodified] => int
                [duedateformatted] => string
                [cohortscount] => int
                [planscount] => int
                [canmanage] => int
                [canread] => int
                [contextname] => string
                [contextnamenoprefix] => string
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="duedate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="duedateformatted">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="cohortscount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="planscount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmanage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canread">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contextname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="contextnamenoprefix">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_list_user_plans

List a user's learning plans.

Arguments**userid** (Required)

The user ID

General structure

```
int //The user ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

```

General structure

list of (
object {
name string //name
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
userid int //userid
templateid int //templateid
origtemplateid int //origtemplateid
status int //status
duedate int //duedate
reviewerid int //reviewerid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
statusname string //statusname
isbasedontemplate int //isbasedontemplate
canmanage int //canmanage
canrequestreview int //canrequestreview
canreview int //canreview
canbeedited int //canbeedited
isactive int //isactive
isdraft int //isdraft
iscompleted int //iscompleted
isinreview int //isinreview
iswaitingforreview int //iswaitingforreview
isreopnallowed int //isreopnallowed
iscompleteallowed int //iscompleteallowed
isunlinkallowed int //isunlinkallowed
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
isstartreviewallowed int //isstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isapproveallowed int //isapproveallowed
isunapproveallowed int //isunapproveallowed
duedateformatted string //duedateformatted
commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
template object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}

```



```
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
)
```

^

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
        (
            [name] => string
            [description] => string
            [descriptionformat] => int
            [userid] => int
            [templateid] => int
            [origtemplateid] => int
            [status] => int
            [duedate] => int
            [reviewerid] => int
            [id] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
            [statusname] => string
            [isbasedontemplate] => int
            [canmanage] => int
            [canrequestreview] => int
            [canreview] => int
            [canbeedited] => int
            [isactive] => int
            [isdraft] => int
            [iscompleted] => int
            [isinreview] => int
            [iswaitingforreview] => int
            [isreopnallowed] => int
            [iscompleteallowed] => int
            [isunlinkallowed] => int
            [isrequestreviewallowed] => int
            [iscancelreviewrequestallowed] => int
            [issstartreviewallowed] => int
            [isstopreviewallowed] => int
            [isapproveallowed] => int
            [isunapproveallowed] => int
            [duedateformatted] => string
            [commentarea] =>
                Array
                (
                    [component] => string
                    [commentarea] => string
                    [itemid] => int
                    [courseid] => int
                    [contextid] => int
                    [cid] => string
                    [autostart] => int
                    [canpost] => int
                    [canview] => int
                    [count] => int
                    [collapsediconkey] => string
                    [displaytotalcount] => int
                    [displaycancel] => int
                    [fullwidth] => int
                    [linktext] => string
                    [notoggle] => int
                    [template] => string
                    [canpostorhascomments] => int
                )
            [reviewer] =>
                Array
                (
                    [id] => int
                    [email] => string
                    [idnumber] => string
                    [phone1] => string
                    [phone2] => string
                    [department] => string
                    [institution] => string
                    [fullname] => string
                    [identity] => string
                    [profileurl] => string
                    [profileimageurl] => string
                    [profileimageurlsmall] => string
                )
            [template] =>
                Array
                (

```

^

```
[shortname] => string
[description] => string
[descriptionformat] => int
[duedate] => int
[visible] => int
[contextid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="userid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="templateid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="origtemplateid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="status">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="duedate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="reviewerid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="statusname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isbasedontemplate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmanage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canrequestreview">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canreview">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canbeedited">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isactive">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isdraft">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscompleted">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isinreview">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iswaitingforreview">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isreopenallowed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscompleteallowed">
        <VALUE>int</VALUE>
      </KEY>
```



```
</KEY>
<KEY name="isunlinkallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="commentarea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="autostart">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canview">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="collapsediconkey">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="displaytotalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="displaycancel">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullwidth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="linktext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="notoggle">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="template">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="canpostorhascomments">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```
<KEY name="reviewer">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="email">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="identity">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="template">
    <SINGLE>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedateformatted">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="cohortscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="planscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanage">
            <VALUE>int</VALUE>
        </KEY>
```



```

</KEY>
<KEY name="canread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_move_down_competency

Re-order a competency.

Arguments

id (Required)

The competency id



General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_move_up_competency

Re-order a competency.

Arguments

id (Required)

The competency id

**General structure**

```
int //The competency id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_plan_cancel_review_request

Cancel the review of a plan.

Arguments

id (Required)

The plan ID

**General structure**

```
int //The plan ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_plan_request_review

Request for a plan to be reviewed.

Arguments

id (Required)

The plan ID

**General structure**

```
int //The plan ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_plan_start_review

Start the review of a plan.

Arguments

id (Required)

The plan ID

**General structure**

```
int //The plan ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_plan_stop_review

Stop the review of a plan.

Arguments

id (Required)

The plan ID

**General structure**

```
int //The plan ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_read_competency

Load a summary of a competency.

Arguments

id (Required)

Data base record id for the competency

**General structure**

```
int //Data base record id for the competency
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    sortorder int //sortorder
    parentid int //parentid
    path string //path
    ruleoutcome int //ruleoutcome
    ruletype string //ruletype
    ruleconfig string //ruleconfig
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    competencyframeworkid int //competencyframeworkid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
}
```

XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [idnumber] => string
    [description] => string
    [descriptionformat] => int
    [sortorder] => int
    [parentid] => int
    [path] => string
    [ruleoutcome] => int
    [ruletype] => string
    [ruleconfig] => string
    [scaleid] => int
    [scaleconfiguration] => string
    [competencyframeworkid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="sortorder">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="parentid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="path">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="ruleoutcome">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ruletype">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="ruleconfig">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="scaleid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleconfiguration">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="competencyframeworkid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_read_competency_framework

Load a summary of a competency framework.

Arguments**id** (Required)

Data base record id for the framework

General structure

```
int //Data base record id for the framework
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response**General structure**

```
object {
    shortname string //shortname
    idnumber string //idnumber
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    visible int //visible
    scaleid int //scaleid
    scaleconfiguration string //scaleconfiguration
    contextid int //contextid
    taxonomies string //taxonomies
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    canmanage int //canmanage
    competenciescount int //competenciescount
    contextname string //contextname
    contextnamenoprefix string //contextnamenoprefix
}
```



XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [idnumber] => string
    [description] => string
    [descriptionformat] => int
    [visible] => int
    [scaleid] => int
    [scaleconfiguration] => string
    [contextid] => int
    [taxonomies] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [canmanage] => int
    [competenciescount] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="scaleconfiguration">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="taxonomies">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="competenciescount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_read_plan

Load a learning plan.

Arguments**id** (Required)

Data base record id for the plan

General structure

```
int //Data base record id for the plan
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```

object {
    name string //name
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    userid int //userid
    templateid int //templateid
    origtemplateid int //origtemplateid
    status int //status
    duedate int //duedate
    reviewerid int //reviewerid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    statusname string //statusname
    isbasedontemplate int //isbasedontemplate
    canmanage int //canmanage
    canrequestreview int //canrequestreview
    canreview int //canreview
    canbeedited int //canbeedited
    isactive int //isactive
    isdraft int //isdraft
    iscompleted int //iscompleted
    isinreview int //isinreview
    iswaitingforreview int //iswaitingforreview
    isreopenallowed int //isreopenallowed
    incompleteallowed int //incompleteallowed
    isunlinkallowed int //isunlinkallowed
    isrequestreviewallowed int //isrequestreviewallowed
    iscancelreviewrequestallowed int //iscancelreviewrequestallowed
    isstartreviewallowed int //isstartreviewallowed
    isstopreviewallowed int //isstopreviewallowed
    isapprovewallowed int //isapprovewallowed
    isunapprovewallowed int //isunapprovewallowed
    duedateformatted string //duedateformatted
    commentarea object {
        component string //component
        commentarea string //commentarea
    }
    itemid int //itemid
    courseid int //courseid
    contextid int //contextid
    cid string //cid
    autostart int //autostart
    canpost int //canpost
    canview int //canview
    count int //count
    collapsediconkey string //collapsediconkey
    displaytotalcount int //displaytotalcount
    displaycancel int //displaycancel
    fullwidth int //fullwidth
    linktext string //linktext
    notoggle int //notoggle
    template string //template
    canpostorhascomments int //canpostorhascomments
}
reviewer object {
    id int //id
    email string //email
    idnumber string //idnumber
    phone1 string //phone1
    phone2 string //phone2
    department string //department
    institution string //institution
    fullname string //fullname
    identity string //identity
    profileurl string //profileurl
    profileimageurl string //profileimageurl
    profileimageurlsmall string //profileimageurlsmall
}
template object {
    shortname string //shortname
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    duedate int //duedate
    visible int //visible
    contextid int //contextid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    duedateformatted string //duedateformatted
}

```



```
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [name] => string
    [description] => string
    [descriptionformat] => int
    [userid] => int
    [templateid] => int
    [origtemplateid] => int
    [status] => int
    [duedate] => int
    [reviewerid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [statusname] => string
    [isbasedontemplate] => int
    [canmanage] => int
    [canrequestreview] => int
    [canreview] => int
    [canbeedited] => int
    [isactive] => int
    [isdraft] => int
    [iscompleted] => int
    [isinreview] => int
    [iswaitingforreview] => int
    [isreopenallowed] => int
    [iscompleteallowed] => int
    [isunlinkallowed] => int
    [isrequestreviewallowed] => int
    [iscancelreviewrequestallowed] => int
    [isstartreviewallowed] => int
    [isstopreviewallowed] => int
    [isapproveallowed] => int
    [isunapproveallowed] => int
    [duedateformatted] => string
    [commentarea] =>
        Array
        (
            [component] => string
            [commentarea] => string
            [itemid] => int
            [courseid] => int
            [contextid] => int
            [cid] => string
            [autostart] => int
            [canpost] => int
            [canview] => int
            [count] => int
            [collapsediconkey] => string
            [displaytotalcount] => int
            [displaycancel] => int
            [fullwidth] => int
            [linktext] => string
            [notoggle] => int
            [template] => string
            [canpostorhascomments] => int
        )
    [reviewer] =>
        Array
        (
            [id] => int
            [email] => string
            [idnumber] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [fullname] => string
            [identity] => string
            [profileurl] => string
            [profileimageurl] => string
            [profileimageurlsmall] => string
        )
    [template] =>
        Array
        (
            [shortname] => string
            [description] => string
            [descriptionformat] => int
        )
)

```



```
[duedate] => int
[visible] => int
[contextid] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="templateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="origtemplateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="reviewerid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="isbasedontemplate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canrequestreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canbeedited">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isactive">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isdraft">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isinreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iswaitingforreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isreopenallowed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleteallowed">
      <VALUE>int</VALUE>
    </KEY>
```



```
<KEY name="isunlinkallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="commentarea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="autostart">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canview">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="collapsediconkey">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="displaytotalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="displaycancel">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullwidth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="linktext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="notoggle">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="template">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="canpostorhascomments">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="reviewer">
```



```
<SINGLE>
    <KEY name="id">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="email">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone1">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone2">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="department">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="institution">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="fullname">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="identity">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</KEY>
<KEY name="template">
    <SINGLE>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedateformatted">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="cohortscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="planscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanage">
            <VALUE>int</VALUE>
        </KEY>
```



```

<KEY name="canread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_read_template

Load a summary of a learning plan template.

Arguments

id (Required)

Data base record id for the template

**General structure**

```
int //Data base record id for the template
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```
object {
    shortname string //shortname
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    duedate int //duedate
    visible int //visible
    contextid int //contextid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    duedateformatted string //duedateformatted
    cohortscount int //cohortscount
    planscount int //planscount
    canmanage int //canmanage
    canread int //canread
    contextname string //contextname
    contextnamenoprefix string //contextnamenoprefix
}
```

XML-RPC (PHP structure)

```
Array
(
    [shortname] => string
    [description] => string
    [descriptionformat] => int
    [duedate] => int
    [visible] => int
    [contextid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [duedateformatted] => string
    [cohortscount] => int
    [planscount] => int
    [canmanage] => int
    [canread] => int
    [contextname] => string
    [contextnamenoprefix] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="shortname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedateformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="cohortscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="planscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="contextname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextnamenoprefix">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_read_user_evidence

Read an evidence of prior learning.

Arguments**id** (Required)

The user evidence ID.

General structure

```
int //The user evidence ID.
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

General structure

```
object {
    userid int    //userid
    name string   //name
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    url string   //url
    id int       //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    canmanage int //canmanage
    competencycount int //competencycount
    competencies //competencies
    list of (
        object {
            shortname string //shortname
            idnumber string //idnumber
            description string //description
            descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
            sortorder int //sortorder
            parentid int //parentid
            path string //path
            ruleoutcome int //ruleoutcome
            rulename string //rulename
            ruleconfig string //ruleconfig
            scaleid int //scaleid
            scaleconfiguration string //scaleconfiguration
            competencyframeworkid int //competencyframeworkid
            id int //id
            timecreated int //timecreated
            timemodified int //timemodified
            usermodified int //usermodified
        }
    )filecount int //filecount
    files //files
    list of (
        object {
            contextid int //contextid
            component string //component
            filearea string //filearea
            itemid int //itemid
            filepath string //filepath
            filename string //filename
            isdir int //isdir
            isimage int //isimage
            timemodified int //timemodified
            timecreated int //timecreated
            filesize int //filesize
            author string //author
            license string //license
            filenameshort string //filenameshort
            filesizeformatted string //filesizeformatted
            icon string //icon
            timecreatedformatted string //timecreatedformatted
            timemodifiedformatted string //timemodifiedformatted
            url string //url
        }
    )hasurlorfiles int //hasurlorfiles
    urlshort string //urlshort
}
```



XML-RPC (PHP structure)

```

Array
(
    [userid] => int
    [name] => string
    [description] => string
    [descriptionformat] => int
    [url] => string
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [canmanage] => int
    [competencycount] => int
    [competencies] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [shortname] => string
                            [idnumber] => string
                            [description] => string
                            [descriptionformat] => int
                            [sortorder] => int
                            [parentid] => int
                            [path] => string
                            [ruleoutcome] => int
                            [ruletype] => string
                            [ruleconfig] => string
                            [scaleid] => int
                            [scaleconfiguration] => string
                            [competencyframeworkid] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                        )
                )
            )
    [filecount] => int
    [files] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [contextid] => int
                            [component] => string
                            [filearea] => string
                            [itemid] => int
                            [filepath] => string
                            [filename] => string
                            [isdir] => int
                            [isimage] => int
                            [timemodified] => int
                            [timecreated] => int
                            [filesize] => int
                            [author] => string
                            [license] => string
                            [filenameshort] => string
                            [filesizeformatted] => string
                            [icon] => string
                            [timecreatedformatted] => string
                            [timemodifiedformatted] => string
                            [url] => string
                        )
                )
            )
    [hasurlorfiles] => int
    [urlshort] => string
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="userid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="competencycount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="competencies">
      <MULTIPLE>
        <SINGLE>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="description">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="descriptionformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="parentid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="path">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleoutcome">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="ruletype">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="ruleconfig">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="scaleid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="scaleconfiguration">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="competencyframeworkid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
```



```
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="filecount">
    <VALUE>int</VALUE>
</KEY>
</KEY>
<KEY name="files">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodifiedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="hasurlorfiles">
    <VALUE>int</VALUE>
</KEY>
<KEY name="urlshort">
    <VALUE>string</VALUE>

```



```
</KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_remove_competency_from_course`

Remove a competency from a course

Arguments**courseid** (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_remove_competency_from_plan`

Remove the competency from a learning plan



Arguments

`planid` (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

`competencyid` (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_remove_competency_from_template`

Remove a competency from a template

Arguments

templateid (Required)

The template id

General structure

```
int //The template id
```

XML-RPC (PHP structure)

```
[templateid] => int
```

REST (POST parameters)

```
templateid= int
```

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_remove_related_competency`

Remove a related competency

Arguments

`competencyid` (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

`relatedcompetencyid` (Required)

The related competency id

General structure

```
int //The related competency id
```

XML-RPC (PHP structure)

```
[relatedcompetencyid] => int
```

REST (POST parameters)

```
relatedcompetencyid= int
```



Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_reopen_plan

Reopen learning plan.

Arguments**planid** (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

**REST (POST parameters)**

```
planid= int
```

Response

True if reopening learning plan was successful

General structure

```
int //True if reopening Learning plan was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_reorder_course_competency

Move a course competency to a new relative sort order.

Arguments**courseid** (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

**REST (POST parameters)**

```
courseid= int
```

competencyidfrom (Required)

The competency id we are moving

General structure

```
int //The competency id we are moving
```

XML-RPC (PHP structure)

```
[competencyidfrom] => int
```

REST (POST parameters)

```
competencyidfrom= int
```

competencyidto (Required)

The competency id we are moving to

General structure

```
int //The competency id we are moving to
```

XML-RPC (PHP structure)

```
[competencyidto] => int
```

REST (POST parameters)

```
competencyidto= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_competency_reorder_plan_competency`

Move a plan competency to a new relative sort order.

Arguments

planid (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

competencyidfrom (Required)

The competency id we are moving

General structure

```
int //The competency id we are moving
```

XML-RPC (PHP structure)

```
[competencyidfrom] => int
```

REST (POST parameters)

```
competencyidfrom= int
```

competencyidto (Required)

The competency id we are moving to

General structure

```
int //The competency id we are moving to
```



XML-RPC (PHP structure)

```
[competencyidto] => int
```

REST (POST parameters)

```
competencyidto= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_reorder_template_competency

Move a template competency to a new relative sort order.

Arguments**templateid** (Required)

The template id

General structure

```
int //The template id
```

**XML-RPC (PHP structure)**

```
[templateid] => int
```

REST (POST parameters)

```
templateid= int
```

competencyidfrom (Required)

The competency id we are moving

General structure

```
int //The competency id we are moving
```

XML-RPC (PHP structure)

```
[competencyidfrom] => int
```

REST (POST parameters)

```
competencyidfrom= int
```

competencyidto (Required)

The competency id we are moving to

General structure

```
int //The competency id we are moving to
```

XML-RPC (PHP structure)

```
[competencyidto] => int
```

REST (POST parameters)

```
competencyidto= int
```

Response

True if successful.

General structure

```
int //True if successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

[core_competency_request_review_of_user_evidence_linked_competencies](#)

Send user evidence competencies in review

Arguments**id** (Required)

The user evidence ID.

General structure

```
int //The user evidence ID.
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if all competencies were send to review

General structure

```
int //True if all competencies were send to review
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_search_competencies

Search a list of a competencies.

Arguments

searchtext (Required)

Text to search for

General structure

```
string //Text to search for
```

XML-RPC (PHP structure)

```
[searchtext] => string
```

REST (POST parameters)

```
searchtext= string
```

competencyframeworkid (Required)

Competency framework id

General structure

```
int //Competency framework id
```

XML-RPC (PHP structure)

```
[competencyframeworkid] => int
```

REST (POST parameters)

```
competencyframeworkid= int
```

Response**General structure**

```
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
})
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [shortname] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [sortorder] => int
                [parentid] => int
                [path] => string
                [ruleoutcome] => int
                [ruletype] => string
                [ruleconfig] => string
                [scaleid] => int
                [scaleconfiguration] => string
                [competencyframeworkid] => int
                [id] => int
                [timecreated] => int
                [timemodified] => int
                [usermodified] => int
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="sortorder">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="parentid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="path">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleoutcome">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="ruletype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="ruleconfig">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="scaleid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scaleconfiguration">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="competencyframeworkid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_set_course_competency_ruleoutcome

Modify the ruleoutcome value for course competency

Arguments**coursecompetencyid** (Required)

Data base record id for the course competency

General structure

```
int //Data base record id for the course competency
```

XML-RPC (PHP structure)

```
[coursecompetencyid] => int
```

REST (POST parameters)

```
coursecompetencyid= int
```

ruleoutcome (Required)

Ruleoutcome value

General structure

```
int //Ruleoutcome value
```

XML-RPC (PHP structure)

```
[ruleoutcome] => int
```

REST (POST parameters)

```
ruleoutcome= int
```

**Response**

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_set_parent_competency

Set a new parent for a competency.

Arguments

competencyid (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

parentid (Required)

The new competency parent id



General structure

```
int //The new competency parent id
```

XML-RPC (PHP structure)

```
[parentid] => int
```

REST (POST parameters)

```
parentid= int
```

Response

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_template_has_related_data

Check if a template has related data

Arguments

id (Required)

The template id

**General structure**

```
int //The template id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the template has related data

General structure

```
int //True if the template has related data
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_template_viewed

Log event template viewed

Arguments

id (Required)

Data base record id for the template

**General structure**

```
int //Data base record id for the template
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

True if the log of the view was successful

General structure

```
int //True if the log of the view was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_unapprove_plan

Unapprove a plan.

Arguments

id (Required)

The plan ID

**General structure**

```
int //The plan ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_unlink_plan_from_template

Unlink a plan form it template.

Arguments**planid** (Required)

Data base record id for the plan

**General structure**

```
int //Data base record id for the plan
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

Response

True if the unlink was successful

General structure

```
int //True if the unlink was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_update_competency

Update a competency.

Arguments

competency (Required)



General structure

```

object {
    shortname string Optional //shortname
    idnumber string Optional //idnumber
    description string Optional //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    sortorder int Optional //sortorder
    parentid int Optional //parentid
    path string Optional //path
    ruleoutcome int Optional //ruleoutcome
    ruletype string Optional //ruletype
    ruleconfig string Optional //ruleconfig
    scaleid int Optional //scaleid
    scaleconfiguration string Optional //scaleconfiguration
    competencyframeworkid int Optional //competencyframeworkid
    id int //id
    timecreated int Optional //timecreated
    timemodified int Optional //timemodified
    usermodified int Optional //usermodified
}

```

XML-RPC (PHP structure)

```

[competency] =>
    Array
        (
            [shortname] => string
            [idnumber] => string
            [description] => string
            [descriptionformat] => int
            [sortorder] => int
            [parentid] => int
            [path] => string
            [ruleoutcome] => int
            [ruletype] => string
            [ruleconfig] => string
            [scaleid] => int
            [scaleconfiguration] => string
            [competencyframeworkid] => int
            [id] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )

```

REST (POST parameters)

```

competency[shortname]= string
competency[idnumber]= string
competency[description]= string
competency[descriptionformat]= int
competency[sortorder]= int
competency[parentid]= int
competency[path]= string
competency[ruleoutcome]= int
competency[ruletype]= string
competency[ruleconfig]= string
competency[scaleid]= int
competency[scaleconfiguration]= string
competency[competencyframeworkid]= int
competency[id]= int
competency[timecreated]= int
competency[timemodified]= int
competency[usermodified]= int

```

**Response**

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_update_competency_framework

Update a competency framework.

Arguments

competencyframework (Required)

**General structure**

```
object {
    shortname string Optional //shortname
    idnumber string Optional //idnumber
    description string Optional //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    visible int Optional //visible
    scaleid int Optional //scaleid
    scaleconfiguration string Optional //scaleconfiguration
    contextid int Optional //The context id
    contextlevel string Optional //The context Level
    instanceid int Optional //The Instance id
    taxonomies string Optional //taxonomies
    id int id
    timecreated int Optional //timecreated
    timemodified int Optional //timemodified
    usermodified int Optional //usermodified
}
```

XML-RPC (PHP structure)

```
[competencyframework] =>
    Array
        (
            [shortname] => string
            [idnumber] => string
            [description] => string
            [descriptionformat] => int
            [visible] => int
            [scaleid] => int
            [scaleconfiguration] => string
            [contextid] => int
            [contextlevel] => string
            [instanceid] => int
            [taxonomies] => string
            [id] => int
            [timecreated] => int
            [timemodified] => int
            [usermodified] => int
        )
    )
```

REST (POST parameters)

```
competencyframework[shortname]= string
competencyframework[idnumber]= string
competencyframework[description]= string
competencyframework[descriptionformat]= int
competencyframework[visible]= int
competencyframework[scaleid]= int
competencyframework[scaleconfiguration]= string
competencyframework[contextid]= int
competencyframework[contextlevel]= string
competencyframework[instanceid]= int
competencyframework[taxonomies]= string
competencyframework[id]= int
competencyframework[timecreated]= int
competencyframework[timemodified]= int
competencyframework[usermodified]= int
```

Response

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_update_course_competency_settings

Update the course competency settings

Arguments**courseid** (Required)

Course id for the course to update

General structure

```
int //Course id for the course to update
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

settings (Required)**General structure**

```
object {
pushratingstouserplans int //New value of the setting
}
```

**XML-RPC (PHP structure)**

```
[settings] =>
  Array
  (
    [pushratingstouserplans] => int
  )
```

REST (POST parameters)

```
settings[pushratingstouserplans]= int
```

Response

True if the update was successful.

General structure

```
int //True if the update was successful.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_update_plan

Updates a learning plan.

Arguments

plan (Required) 

General structure

```
object {
    name string Optional //name
    description string Optional //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    userid int Optional //userid
    templateid int Optional //templateid
    origtemplateid int Optional //origtemplateid
    status int Optional //status
    duedate int Optional //duedate
    reviewerid int Optional //reviewerid
    id int //id
    timecreated int Optional //timecreated
    timemodified int Optional //timemodified
    usermodified int Optional //usermodified
}
```

XML-RPC (PHP structure)

```
[plan] =>
  Array
  (
    [name] => string
    [description] => string
    [descriptionformat] => int
    [userid] => int
    [templateid] => int
    [origtemplateid] => int
    [status] => int
    [duedate] => int
    [reviewerid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
  )
```

REST (POST parameters)

```
plan[name]= string
plan[description]= string
plan[descriptionformat]= int
plan[userid]= int
plan[templateid]= int
plan[origtemplateid]= int
plan[status]= int
plan[duedate]= int
plan[reviewerid]= int
plan[id]= int
plan[timecreated]= int
plan[timemodified]= int
plan[usermodified]= int
```

Response

General structure

```

object {
    name string //name
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    userid int //userid
    templateid int //templateid
    origtemplateid int //origtemplateid
    status int //status
    duedate int //duedate
    reviewerid int //reviewerid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    statusname string //statusname
    isbasedontemplate int //isbasedontemplate
    canmanage int //canmanage
    canrequestreview int //canrequestreview
    canreview int //canreview
    canbeedited int //canbeedited
    isactive int //isactive
    isdraft int //isdraft
    iscompleted int //iscompleted
    isinreview int //isinreview
    iswaitingforreview int //iswaitingforreview
    isreopenallowed int //isreopenallowed
    incompleteallowed int //incompleteallowed
    isunlinkallowed int //isunlinkallowed
    isrequestreviewallowed int //isrequestreviewallowed
    iscancelreviewrequestallowed int //iscancelreviewrequestallowed
    isstartreviewallowed int //isstartreviewallowed
    isstopreviewallowed int //isstopreviewallowed
    isapprovewallowed int //isapprovewallowed
    isunapprovewallowed int //isunapprovewallowed
    duedateformatted string //duedateformatted
    commentarea object {
        component string //component
        commentarea string //commentarea
    }
    itemid int //itemid
    courseid int //courseid
    contextid int //contextid
    cid string //cid
    autostart int //autostart
    canpost int //canpost
    canview int //canview
    count int //count
    collapsediconkey string //collapsediconkey
    displaytotalcount int //displaytotalcount
    displaycancel int //displaycancel
    fullwidth int //fullwidth
    linktext string //linktext
    notoggle int //notoggle
    template string //template
    canpostorhascomments int //canpostorhascomments
}
reviewer object {
    id int //id
    email string //email
    idnumber string //idnumber
    phone1 string //phone1
    phone2 string //phone2
    department string //department
    institution string //institution
    fullname string //fullname
    identity string //identity
    profileurl string //profileurl
    profileimageurl string //profileimageurl
    profileimageurlsmall string //profileimageurlsmall
}
template object {
    shortname string //shortname
    description string //description
    descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    duedate int //duedate
    visible int //visible
    contextid int //contextid
    id int //id
    timecreated int //timecreated
    timemodified int //timemodified
    usermodified int //usermodified
    duedateformatted string //duedateformatted
}

```



```
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
```

^

XML-RPC (PHP structure)

```

Array
(
    [name] => string
    [description] => string
    [descriptionformat] => int
    [userid] => int
    [templateid] => int
    [origtemplateid] => int
    [status] => int
    [duedate] => int
    [reviewerid] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [statusname] => string
    [isbasedontemplate] => int
    [canmanage] => int
    [canrequestreview] => int
    [canreview] => int
    [canbeedited] => int
    [isactive] => int
    [isdraft] => int
    [iscompleted] => int
    [isinreview] => int
    [iswaitingforreview] => int
    [isreopenallowed] => int
    [iscompleteallowed] => int
    [isunlinkallowed] => int
    [isrequestreviewallowed] => int
    [iscancelreviewrequestallowed] => int
    [isstartreviewallowed] => int
    [isstopreviewallowed] => int
    [isapproveallowed] => int
    [isunapproveallowed] => int
    [duedateformatted] => string
    [commentarea] =>
        Array
        (
            [component] => string
            [commentarea] => string
            [itemid] => int
            [courseid] => int
            [contextid] => int
            [cid] => string
            [autostart] => int
            [canpost] => int
            [canview] => int
            [count] => int
            [collapsediconkey] => string
            [displaytotalcount] => int
            [displaycancel] => int
            [fullwidth] => int
            [linktext] => string
            [notoggle] => int
            [template] => string
            [canpostorhascomments] => int
        )
    [reviewer] =>
        Array
        (
            [id] => int
            [email] => string
            [idnumber] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [fullname] => string
            [identity] => string
            [profileurl] => string
            [profileimageurl] => string
            [profileimageurlsmall] => string
        )
    [template] =>
        Array
        (
            [shortname] => string
            [description] => string
            [descriptionformat] => int
        )
)

```



```
[duedate] => int
[visible] => int
[contextid] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="description">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="descriptionformat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="templateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="origtemplateid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="reviewerid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="isbasedontemplate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canmanage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canrequestreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canbeedited">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isactive">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isdraft">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isinreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iswaitingforreview">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isreopenallowed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="iscompleteallowed">
      <VALUE>int</VALUE>
    </KEY>
```



```
<KEY name="isunlinkallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="commentarea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="autostart">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canview">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="collapsediconkey">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="displaytotalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="displaycancel">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullwidth">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="linktext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="notoggle">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="template">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="canpostorhascomments">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="reviewer">
```



```
<SINGLE>
    <KEY name="id">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="email">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="idnumber">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone1">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="phone2">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="department">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="institution">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="fullname">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="identity">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</KEY>
<KEY name="template">
    <SINGLE>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="description">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="descriptionformat">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usermodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="duedateformatted">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="cohortscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="planscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanage">
            <VALUE>int</VALUE>
        </KEY>
```



```

<KEY name="canread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_update_template

Update a learning plan template.

Arguments

template (Required)



General structure

```

object {
shortname string Optional //shortname
description string Optional //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int Optional //duedate
visible int Optional //visible
contextid int Optional //The context id
contextlevel string Optional //The context Level
instanceid int Optional //The Instance id
id int //id
timecreated int Optional //timecreated
timemodified int Optional //timemodified
usermodified int Optional //usermodified
}

```

XML-RPC (PHP structure)

```
[template] =>
    Array
    (
        [shortname] => string
        [description] => string
        [descriptionformat] => int
        [duedate] => int
        [visible] => int
        [contextid] => int
        [contextlevel] => string
        [instanceid] => int
        [id] => int
        [timecreated] => int
        [timemodified] => int
        [usermodified] => int
    )
```

REST (POST parameters)

```
template[shortname]= string
template[description]= string
template[descriptionformat]= int
template[duedate]= int
template[visible]= int
template[contextid]= int
template[contextlevel]= string
template[instanceid]= int
template[id]= int
template[timecreated]= int
template[timemodified]= int
template[usermodified]= int
```

Response

True if the update was successful

General structure

```
int //True if the update was successful
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_user_competency_cancel_review_request

Cancel a review request.

Arguments**userid** (Required)

The user ID

General structure

```
int //The user ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

competencyid (Required)

The competency ID

General structure

```
int //The competency ID
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

**Response**

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_user_competency_plan_viewed

Log the user competency plan viewed event.

Arguments**competencyid** (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

userid (Required)

The user id

**General structure**

```
int //The user id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

planid (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

Response

True if the event user competency plan viewed was logged

General structure

```
int //True if the event user competency plan viewed was Logged
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_competency_user_competency_request_review

Request a review.

Arguments

userid (Required)

The user ID

General structure

```
int //The user ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

competencyid (Required)

The competency ID

General structure

```
int //The competency ID
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_user_competency_start_review

Start a review.

Arguments

userid (Required)

The user ID

General structure

```
int //The user ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

competencyid (Required)

The competency ID

General structure

```
int //The competency ID
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```



Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_user_competency_stop_review

Stop a review.

Arguments**userid** (Required)

The user ID

General structure

```
int //The user ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

competencyid (Required)

The competency ID

General structure

```
int //The competency ID
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

Response

The success

General structure

```
int //The success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes



core_competency_user_competency_viewed

Log the user competency viewed event.

Arguments

usercompetencyid (Required)

The user competency id

General structure

```
int //The user competency id
```

XML-RPC (PHP structure)

```
[usercompetencyid] => int
```

REST (POST parameters)

```
usercompetencyid= int
```

Response

True if the event user competency viewed was logged

General structure

```
int //True if the event user competency viewed was logged
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes



`core_competency_user_competency_viewed_in_course`

Log the user competency viewed in course event

Arguments

`competencyid` (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

userid (Required)

The user id

General structure

```
int //The user id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

courseid (Required)

The course id

General structure

```
int //The course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

True if the event user competency viewed in course was logged

General structure

```
int //True if the event user competency viewed in course was Logged
```

**XML-RPC (PHP structure)**

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_competency_user_competency_viewed_in_plan

Log the user competency viewed in plan event.

Arguments**competencyid** (Required)

The competency id

General structure

```
int //The competency id
```

XML-RPC (PHP structure)

```
[competencyid] => int
```

REST (POST parameters)

```
competencyid= int
```

userid (Required)

The user id

General structure

```
int //The user id
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

planid (Required)

The plan id

General structure

```
int //The plan id
```

XML-RPC (PHP structure)

```
[planid] => int
```

REST (POST parameters)

```
planid= int
```

Response

True if the event user competency viewed in plan was logged

General structure

```
int //True if the event user competency viewed in plan was Logged
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_completion_get_activities_completion_status

Return the activities completion status for a user in a course.

Arguments

courseid (Required)

Course ID

General structure

```
int //Course ID
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Required)

User ID

General structure

```
int //User ID
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response



```
General structure

object {
    statuses //List of activities status
    list of (
        //Activity
        object {
            cmid int //course module ID
            modname string //activity module name
            instance int //instance ID
            state int //Completion state value:
                0 means incomplete,
                1 complete,
                2 complete pass,
                3 complete fail
            timecompleted int //timestamp for completed activity
            tracking int //type of tracking:
                0 means none,
                1 manual,
                2 automatic
            overrideby int Optional //The user id who has overriden the status, or null
            valueused int Optional //Whether the completion status affects the availability of another activity.
            hascompletion int Optional //Whether this activity module has completion enabled
            isautomatic int Optional //Whether this activity module instance tracks completion automatically.
            istrackeduser int Optional //Whether completion is being tracked for this user.
            uservisible int Optional //Whether this activity is visible to the user.
            details Default to "Array
            (
            )
        }
        " //Completion status details
        list of (
            object {
                rulename string //Rule name
                rulevalue object {
                    status int //Completion status
                    description string //Completion description
                }
            }
        )))
    )warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )))
}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [statuses] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [cmid] => int
                                [modname] => string
                                [instance] => int
                                [state] => int
                                [timecompleted] => int
                                [tracking] => int
                                [overrideby] => int
                                [valueused] => int
                                [hascompletion] => int
                                [isautomatic] => int
                                [istrackeduser] => int
                                [uservisible] => int
                                [details] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [rulename] => string
                                                        [rulevalue] =>
                                                            Array
                                                            (
                                                                [
                                                                    [status] => int
                                                                    [description] => string
                                                                )
                                                            )
                                    )
                                )
                            )
                        )
                )
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="statuses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="cmid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="modname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instance">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="state">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecompleted">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="tracking">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="overrideby">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="valueused">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="hascompletion">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isautomatic">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="istrackeduser">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="uservisible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="details">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="rulename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="rulevalue">
                                    <SINGLE>
                                        <KEY name="status">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="description">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```



```
</KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_completion_get_course_completion_status

Returns course completion status.

Arguments**courseid** (Required)

Course ID

General structure

int //Course ID

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

userid (Required)

User ID

General structure

int //User ID

XML-RPC (PHP structure)

[userid] => int

REST (POST parameters)

userid= int

Response

Course completion status

```
General structure

//Course completion status
object {
completionstatus //Course status
object {
completed int //true if the course is complete, false otherwise
aggregation int //aggregation method 1 means all, 2 means any
completions list of (
    //Completions
object {
type int //Completion criteria type
title string //Completion criteria Title
status string //Completion status (Yes/No) a % or number
complete int //Completion status (true/false)
timecompleted int //Timestamp for criteria completetion
details //details
object {
type string //Type description
criteria string //Criteria description
requirement string //Requirement description
status string //Status description, can be anything
}
}
)
})
warnings Optional //list of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```



XML-RPC (PHP structure)

```
Array
(
    [completionstatus] =>
        Array
            (
                [completed] => int
                [aggregation] => int
                [completions] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => int
                                        [title] => string
                                        [status] => string
                                        [complete] => int
                                        [timecompleted] => int
                                        [details] =>
                                            Array
                                                (
                                                    [type] => string
                                                    [criteria] => string
                                                    [requirement] => string
                                                    [status] => string
                                                )
                                    )
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="completionstatus">
            <SINGLE>
                <KEY name="completed">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="aggregation">
                    <VALUE>int</VALUE>
                </KEY>
            <KEY name="completions">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="type">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="title">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="status">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="complete">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="timecompleted">
                            <VALUE>int</VALUE>
                        </KEY>
                    <KEY name="details">
                        <SINGLE>
                            <KEY name="type">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="criteria">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="requirement">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="status">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_completion_mark_course_self_completed

Update the course completion status for the current user (if course self-completion is enabled).

Arguments**courseid** (Required)

Course ID

General structure

int //Course ID

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

Response**General structure**

```
object {
    status int //status, true if success
    warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core completion override activity completion status

Update completion status for a user in an activity by overriding it.

Arguments

userid (Required)

user id

General structure

```
int //user id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

cmid (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

newstate (Required)

the new activity completion state

General structure

```
int //the new activity completion state
```

XML-RPC (PHP structure)

```
[newstate] => int
```

REST (POST parameters)

```
newstate= int
```

Response

General structure

```
object {
cmid int //The course module id
userid int //The user id to which the completion info belongs
state int //The current completion state.
timecompleted int //time of completion
overrideby int //The user id who has overriden the status, or null
tracking int //type of tracking:
                                0 means none, 1 manual, 2 automatic
}
```

XML-RPC (PHP structure)

```
Array
(
    [cmid] => int
    [userid] => int
    [state] => int
    [timecompleted] => int
    [overrideby] => int
    [tracking] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="cmid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="state">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecompleted">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="overrideby">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="tracking">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_completion_update_activity_completion_status_manually

Update completion status for the current user in an activity, only for activities with manual tracking.

Arguments**cmid** (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

completed (Required)

activity completed or not

General structure

```
int //activity completed or not
```

XML-RPC (PHP structure)

```
[completed] => int
```

REST (POST parameters)

```
completed= int
```

Response**General structure**

```
object {
    status int //status, true if success
    warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core contentbank delete content

Delete a content from the content bank.

Arguments

contentids (Required)

General structure

```
list of (
int //The content id to delete
)
```

XML-RPC (PHP structure)

```
[contentids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
contentids[0]= int
```

Response

General structure

```
object {
result int //The processing result
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
[result] => int
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

**core_contentbank_rename_content**

Rename a content in the content bank.

Arguments**contentid** (Required)

The content id to rename

General structure

```
int //The content id to rename
```

XML-RPC (PHP structure)

```
[contentid] => int
```

REST (POST parameters)

```
contentid= int
```

name (Required)

The new name for the content

General structure

```
string //The new name for the content
```

XML-RPC (PHP structure)

```
[name] => string
```

REST (POST parameters)

```
name= string
```

Response**General structure**

```
object {
    result int //The processing result
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [result] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

**core_contentbank_set_content_visibility**

Set the visibility of a content in the content bank.

Arguments**contentid** (Required)

The content id to rename

General structure

```
int //The content id to rename
```

XML-RPC (PHP structure)

```
[contentid] => int
```

REST (POST parameters)

```
contentid= int
```

visibility (Required)

The new visibility for the content

General structure

```
int //The new visibility for the content
```

XML-RPC (PHP structure)

```
[visibility] => int
```

REST (POST parameters)

```
visibility= int
```

Response**General structure**

```
object {
  result int //The processing result
  warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
  [result] => int
  [warnings] =>
    Array
    (
      [0] =>
        Array
        (
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

**core_course_add_content_item_to_user_favourites**

Adds a content item (activity, resource or their subtypes) to the favourites for the user.

Arguments**componentname** (Required)

frankenstyle name of the component to which the content item belongs

General structure

```
string //frankenstyle name of the component to which the content item belongs
```

XML-RPC (PHP structure)

```
[componentname] => string
```

REST (POST parameters)

```
componentname= string
```

contentitemid (Required)

id of the content item

General structure

```
int //id of the content item
```

XML-RPC (PHP structure)

```
[contentitemid] => int
```

REST (POST parameters)

```
contentitemid= int
```

Response**General structure**

```
object {
    id int //The id of the content item
    name string //Name of the content item
    title string //The string title of the content item, human readable
    link string //The link to the content item creation page
    icon string //HTML containing the icon for the content item
    help string //HTML description / help for the content item
    archetype string //The archetype of the module exposing the content item
    componentname string //The name of the component exposing the content item
    purpose string //The purpose of the component exposing the content item
    favourite int //Has the user favourited the content item
    legacyitem int //If this item was pulled from the old callback and has no item id.
    recommended int //Has this item been recommended
}
```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [name] => string
    [title] => string
    [link] => string
    [icon] => string
    [help] => string
    [archetype] => string
    [componentname] => string
    [purpose] => string
    [favourite] => int
    [legacyitem] => int
    [recommended] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="title">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="link">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="icon">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="help">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="archetype">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="componentname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="purpose">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="favourite">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="legacyitem">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="recommended">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_course_check_updates

Check if there is updates affecting the user for the given course and contexts.

Arguments

courseid (Required)

Course id to check

General structure

```
int //Course id to check
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

tocheck (Required)

Instances to check

General structure

```
//Instances to check
list of (
object {
contextlevel string //The context Level for the file Location.
id int //Context instance id
since int //Check updates since this time stamp
})
```

Only module supported right now.

XML-RPC (PHP structure)

```
[tocheck] =>
  Array
    (
      [0] =>
        Array
          (
            [contextlevel] => string
            [id] => int
            [since] => int
          )
    )
```

REST (POST parameters)

```
tocheck[0][contextlevel]= string
tocheck[0][id]= int
tocheck[0][since]= int
```



filter (Default to "Array ()")

Check only for updates in these areas

General structure

```
Default to "Array"
(
)
" //Check only for updates in these areas
list of (
string //Area name: configuration, fileareas, completion, ratings, comments,
         gradeitems, outcomes
) 
```

XML-RPC (PHP structure)

```
[filter] =>
    Array
        (
            [0] => string
        )
```

REST (POST parameters)

```
filter[0]= string
```

Response**General structure**

```
object {
instances list of (
object {
contextlevel string //The context level
id int //Instance id
updates list of (
object {
name string //Name of the area updated.
timeupdated int Optional //Last time was updated
itemids Optional //The ids of the items updated
list of (
int //Instance id
)}
)
)
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [instances] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [contextlevel] => string
                                [id] => int
                                [updates] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [name] => string
                                                        [timeupdated] => int
                                                        [itemids] =>
                                                            Array
                                                            (
                                                                [
                                                                    [0] => int
                                                                )
                                                            )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="instances">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="contextlevel">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="updates">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timeupdated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="itemids">
                                    <MULTIPLE>
                                        <VALUE>int</VALUE>
                                    </MULTIPLE>
                                </KEY>
                                </SINGLE>
                            </MULTIPLE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
            <KEY name="warnings">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="item">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="itemid">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="warningcode">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="message">
                            <VALUE>string</VALUE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_create_categories

Create course categories

Arguments

categories (Required)

General structure

```
list of (
object {
name string //new category name
parent int Default to "0" //the parent category id inside which the new category will be created
           - set to 0 for a root category
idnumber string Optional //the new category idnumber
description string Optional //the new category description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
theme string Optional //the new category theme. This option must be enabled on moodle
}
)
```

XML-RPC (PHP structure)

```
[categories] =>
Array
(
[0] =>
Array
(
[name] => string
[parent] => int
[idnumber] => string
[description] => string
[descriptionformat] => int
[theme] => string
)
)
```

REST (POST parameters)

```
categories[0][name]= string
categories[0][parent]= int
categories[0][idnumber]= string
categories[0][description]= string
categories[0][descriptionformat]= int
categories[0][theme]= string
```

Response**General structure**

```
list of (
object {
id int //new category id
name string //new category name
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
                (
                    [
                        [id] => int
                        [name] => string
                    )
                )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

core_course_create_courses

Create new courses

Arguments**courses** (Required)

courses to create

General structure

```
//courses to create
list of (
object {
fullname string //full name
shortname string //course short name
categoryid int //category id
idnumber string Optional //id number
summary string Optional //summary
summaryformat int Default to "1" //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
format string Default to "topics" //course format: weeks, topics, social, site,..
showgrades int Default to "1" //1 if grades are shown, otherwise 0
newsitems int Default to "5" //number of recent items appearing on the course page
startdate int Optional //timestamp when the course start
enddate int Optional //timestamp when the course end
numsections int Optional //(deprecated, use courseformatoptions) number of weeks/topics
maxbytes int Default to "0" //largest size of file that can be uploaded into the course
showreports int Default to "1" //are activity report shown (yes = 1, no =0)
visible int Optional //1: available to student, 0:not available
hiddensections int Optional //(deprecated, use courseformatoptions) How the hidden sections in the course are displayed to students
groupmode int Default to "2" //no group, separate, visible
groupmodeforce int Default to "1" //1: yes, 0: no
defaultgroupingid int Default to "0" //default grouping id
enablecompletion int Optional //Enabled, control via completion and activity settings. Disabled,
not shown in activity settings.
completionnotify int Optional //1: yes 0: no
lang string Optional //forced course language
forcetheme string Optional //name of the force theme
courseformatoptions Optional //additional options for particular course format
list of (
object {
name string //course format option name
value string //course format option value
})
customfields Optional //custom fields for the course
list of (
object {
shortname string //The shortname of the custom field
value string //The value of the custom field
})
})
```



XML-RPC (PHP structure)

```
[courses] =>
    Array
        (
            [
                [0] =>
                    Array
                        (
                            [fullname] => string
                            [shortname] => string
                            [categoryid] => int
                            [idnumber] => string
                            [summary] => string
                            [summaryformat] => int
                            [format] => string
                            [showgrades] => int
                            [newsitems] => int
                            [startdate] => int
                            [enddate] => int
                            [numsections] => int
                            [maxbytes] => int
                            [showreports] => int
                            [visible] => int
                            [hiddensections] => int
                            [groupmode] => int
                            [groupmodeforce] => int
                            [defaultgroupingid] => int
                            [enablecompletion] => int
                            [completionnotify] => int
                            [lang] => string
                            [forcetheme] => string
                            [courseformatoptions] =>
                                Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                    (
                                                        [
                                                            [name] => string
                                                            [value] => string
                                                        )
                                                    )
                                        )
                                    )
                            [customfields] =>
                                Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                    (
                                                        [
                                                            [shortname] => string
                                                            [value] => string
                                                        )
                                                    )
                                        )
                                    )
                            )
                        )
                    )
            )
        )
    )
}
```



REST (POST parameters)

```

courses[0][fullname]= string
courses[0][shortname]= string
courses[0][categoryid]= int
courses[0][idnumber]= string
courses[0][summary]= string
courses[0][summaryformat]= int
courses[0][format]= string
courses[0][showgrades]= int
courses[0][newsitems]= int
courses[0][startdate]= int
courses[0][enddate]= int
courses[0][numsections]= int
courses[0][maxbytes]= int
courses[0][showreports]= int
courses[0][visible]= int
courses[0][hiddensections]= int
courses[0][groupmode]= int
courses[0][groupmodeforce]= int
courses[0][defaultgroupingid]= int
courses[0][enablecompletion]= int
courses[0][completionnotify]= int
courses[0][lang]= string
courses[0][forcetheme]= string
courses[0][courseformatoptions][0][name]= string
courses[0][courseformatoptions][0][value]= string
courses[0][customfields][0][shortname]= string
courses[0][customfields][0][value]= string

```

Response**General structure**

```

list of (
object {
id int //course id
shortname string //short name
}
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
        (
            [id] => int
            [shortname] => string
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>

```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_delete_categories

Delete course categories

Arguments

categories (Required)

General structure

```
list of (
object {
id int //category id to delete
newparent int Optional //the parent category to move the contents to, if specified
recursive int Default to "0" //1: recursively delete all contents inside this
                    category, 0 (default): move contents to newparent or current parent category (except if parent is root)
}
)
```

XML-RPC (PHP structure)

```
[categories] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [newparent] => int
                        [recursive] => int
                    )
        )
    )
```

**REST (POST parameters)**

```
categories[0][id]= int
categories[0][newparent]= int
categories[0][recursive]= int
```

Response**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_delete_courses

Deletes all specified courses

Arguments

courseids (Required)

General structure

```
list of (
int //course ID
)
```

XML-RPC (PHP structure)

```
[courseids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

courseids[0]= int

**Response****General structure**

```
object {
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

Error message**REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_delete_modules

Deletes all specified module instances

Arguments

cmids (Required)

Array of course module IDs

General structure

```
//Array of course module IDs
list of (
int //course module ID
)
```

XML-RPC (PHP structure)

```
[cmids] =>
  Array
  (
    [0] => int
  )
```

REST (POST parameters)

cmids[0]= int

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**core_course_duplicate_course**

Duplicate an existing course (creating a new one).

Arguments**courseid** (Required)

course to duplicate id

General structure

int //course to duplicate id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

```
courseid= int
```

fullname (Required)

duplicated course full name

General structure

```
string //duplicated course full name
```

XML-RPC (PHP structure)

```
[fullname] => string
```

REST (POST parameters)

```
fullname= string
```

shortname (Required)

duplicated course short name

General structure

```
string //duplicated course short name
```

XML-RPC (PHP structure)

```
[shortname] => string
```

REST (POST parameters)

```
shortname= string
```

categoryid (Required)

duplicated course category parent

General structure

```
int //duplicated course category parent
```

**XML-RPC (PHP structure)**

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

visible (Default to "1")

duplicated course visible, default to yes

General structure

```
int Default to "1" //duplicated course visible, default to yes
```

XML-RPC (PHP structure)

```
[visible] => int
```

REST (POST parameters)

```
visible= int
```

options (Default to "Array ()")

Course duplication options

General structure

```
Default to "Array"
(
)
" //Course duplication options
list of (
object {
name string //The backup option name:
                    "activities" (int) Include course activites (default to 1 that is equal to yes),
                    "blocks" (int) Include course blocks (default to 1 that is equal to yes),
                    "filters" (int) Include course filters (default to 1 that is equal to yes),
                    "users" (int) Include users (default to 0 that is equal to no),
                    "enrolments" (int) Include enrolment methods (default to 1 - restore only with users),
                    "role_assignments" (int) Include role assignments (default to 0 that is equal to no),
                    "comments" (int) Include user comments (default to 0 that is equal to no),
                    "userscompletion" (int) Include user course completion information (default to 0 that is equal to no),
                    "logs" (int) Include course logs (default to 0 that is equal to no),
                    "grade_histories" (int) Include histories (default to 0 that is equal to no)
value string //the value for the option 1 (yes) or 0 (no)
}
)
```

XML-RPC (PHP structure)

```
[options] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

**Response****General structure**

```
object {
id int //course id
shortname string //short name
}
```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [shortname] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="shortname">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_edit_module

Performs an action on course module (change visibility, duplicate, delete, etc.)

Arguments

action (Required)

action: hide, show, stealth, duplicate, delete, moveleft, moveright, group...

General structure

```
string //action: hide, show, stealth, duplicate, delete, moveleft, moveright, group...
```

XML-RPC (PHP structure)

```
[action] => string
```

REST (POST parameters)

```
action= string
```

id (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

sectionreturn (Default to "null")

section to return to

General structure

```
int Default to "null" //section to return to
```

XML-RPC (PHP structure)

```
[sectionreturn] => int
```

REST (POST parameters)

```
sectionreturn= int
```

Response

html to replace the current module with

General structure

```
string //html to replace the current module with
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_edit_section

Performs an action on course section (change visibility, set marker, delete)

Arguments

action (Required)

action: hide, show, stealth, setmarker, removemarker

General structure

```
string //action: hide, show, stealth, setmarker, removemarker
```

XML-RPC (PHP structure)

```
[action] => string
```

REST (POST parameters)

```
action= string
```

id (Required)

course section id

General structure

```
int //course section id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

sectionreturn (Default to "null")

section to return to

General structure

```
int Default to "null" //section to return to
```

XML-RPC (PHP structure)

```
[sectionreturn] => int
```

REST (POST parameters)

```
sectionreturn= int
```

Response

Additional data for javascript (JSON-encoded string)

General structure

```
string //Additional data for javascript (JSON-encoded string)
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_activity_chooser_footer

Fetch the data for the activity chooser footer.

Arguments

courseid (Required)

ID of the course

**General structure**

```
int //ID of the course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

sectionid (Required)

ID of the section

General structure

```
int //ID of the section
```

XML-RPC (PHP structure)

```
[sectionid] => int
```

REST (POST parameters)

```
sectionid= int
```

Response**General structure**

```
object {
  footer int //Is a footer being return by this request?
  customfooterjs string Optional //The path to the plugin JS file
  customfootertemplate string Optional //The prerendered footer
  customcarouseltemplate string Optional //Either "" or the prerendered carousel page
}
```

XML-RPC (PHP structure)

```
Array
(
    [footer] => int
    [customfooterjs] => string
    [customfootertemplate] => string
    [customcarouseltemplate] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="footer">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="customfooterjs">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="customfootertemplate">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="customcarouseltemplate">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_categories

Return category details

Arguments

criteria (Default to "Array ()")

criteria

General structure

```
Default to "Array
(
)
" //criteria
list of (
object {
key string //The category column to search, expected keys (value format) are: "id" (int) the category id,"ids" (string) category ids separated by commas,"name" (string) the category name,"parent" (int) the parent category id,"idnumber" (string) category idnumber - user must have 'moodle/category:manage' to search on idnumber,"visible" (int) whether the returned categories must be visible or hidden. If the key is not passed, then the function return all categories that the user can see. - user must have 'moodle/category:manage' or 'moodle/category:viewhiddencategories' to search on visible,"theme" (string) only return the categories having this theme - user must have 'moodle/category:manage' to search on theme
value string //the value to match
}
)
```

XML-RPC (PHP structure)

```
[criteria] =>
    Array
        (
            [0] =>
                Array
                    (
                        [key] => string
                        [value] => string
                    )
        )
    )
```

REST (POST parameters)

```
criteria[0][key]= string
criteria[0][value]= string
```

addsubcategories (Default to "1")

return the sub categories infos (1 - default) otherwise only the category info (0)

General structure

```
int Default to "1" //return the sub categories infos
                    (1 - default) otherwise only the category info (0)
```

XML-RPC (PHP structure)

```
[addsubcategories] => int
```

REST (POST parameters)

addsubcategories= int

Response**General structure**

```

list of (
    //List of categories
object {
id int    //category id
name string //category name
idnumber string Optional //category id number
description string //category description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
parent int   //parent category id
sortorder int //category sorting order
coursecount int //number of courses in this category
visible int  Optional //1: available, 0:not available
visibleold int Optional //1: available, 0:not available
timemodified int Optional //timestamp
depth int    //category depth
path string  //category path
theme string Optional //category theme
}
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [id] => int
                [name] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [parent] => int
                [sortorder] => int
                [coursecount] => int
                [visible] => int
                [visibleold] => int
                [timemodified] => int
                [depth] => int
                [path] => string
                [theme] => string
            )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="parent">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="sortorder">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="coursecount">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="visible">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="visibleold">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="depth">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="path">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="theme">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_contents

Get course contents

Arguments

courseid (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

options (Default to "Array ()")

Options, used since Moodle 2.9

General structure

```
Default to "Array
(
)
" //Options, used since Moodle 2.9
list of (
object {
name string //The expected keys (value format) are:
                    excludemodules (bool) Do not return modules, return only the sections structure
                    excludecontents (bool) Do not return module contents (i.e: files inside a resource)
                    includestealthmodules (bool) Return stealth modules for students in a special
                        section (with id -1)
                    sectionid (int) Return only this section
                    sectionnumber (int) Return only this section with number (order)
                    cmid (int) Return only this module information (among the whole sections structure)
                    modname (string) Return only modules with this name "Label, forum, etc..."
                    modid (int) Return only the module with this id (to be used with modname
value string //the value of the option,
                    this param is personally validated in the external function.
}
)
```

XML-RPC (PHP structure)

```
[options] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response

```

General structure

list of (
object {
id int //Section ID
name string //Section name
visible int Optional //is the section visible
summary string //Section description
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
section int Optional //Section number inside the course
hiddenbynumsections int Optional //Whether a section hidden in the course format
uservisible int Optional //Is the section visible for the user?
availabilityinfo string Optional //Availability information.
modules //List of module
list of (
object {
id int //activity id
url string Optional //activity url
name string //activity module name
instance int Optional //instance id
contextid int Optional //Activity context id.
description string Optional //activity description
visible int Optional //is the module visible
uservisible int Optional //Is the module visible for the user?
availabilityinfo string Optional //Availability information.
visibleoncoursepage int Optional //is the module visible on course page
modicon string //activity icon url
modname string //activity module type
modplural string //activity module plural name
availability string Optional //module availability settings
indent int //number of indentation in the site
onclick string Optional //On click action.
afterlink string Optional //After link info to be displayed.
customdata string Optional //Custom data (JSON encoded).
noviewlink int Optional //Whether the module has no view page
completion int Optional //Type of completion tracking:
                                0 means none, 1 manual, 2 automatic.
completiondata object {
state int //overall completion state of this course module.
timecompleted int //course completion timestamp.
overrideby int //user ID that has overridden the completion state of this activity for the user.
valueused int //True if module is used in a condition, false otherwise.
hascompletion int //Whether this activity module has completion enabled.
isautomatic int //Whether this activity module instance tracks completion automatically.
istrackeduser int //Checks whether completion is being tracked for this user.
uservisible int //Whether this activity is visible to user.
details //An array of completion details containing the description and status.
list of (
object {
rulename string //rule name
rulevalue object {
status int //status
description string //description
}
}
)
downloadcontent int Optional //The download content value
dates Default to "Array
(
)
" //Course dates
list of (
object {
label string //date label
timestamp int //date timestamp
relativeto int Optional //relative date timestamp
dataid string Optional //cm data id
}
)
contents Default to "Array
(
)
" //Course contents
list of (
object {
type string //a file or a folder or external link
filename string //filename
filepath string //filepath
filesize int //filesize
fileurl string Optional //downloadable file url
content string Optional //Raw content, will be used when type is content
timecreated int //Time created
timemodified int //Time modified
sortorder int //Content sort order
}
)

```



```
mimetype string Optional //File mime type.  
isexternalfile int Optional //Whether is an external file.  
repositorytype string Optional //The repository type for external files.  
userid int //User who added this content to moodle  
author string //Content owner  
license string //Content license  
tags Optional //Tags  
list of (  
object {  
id int //Tag id.  
name string //Tag name.  
rawname string //The raw, unnormalised name for the tag as entered by users.  
isstandard int //Whether this tag is standard.  
tagcollid int //Tag collection id.  
taginstanceid int //Tag instance id.  
taginstancecontextid int //Context the tag instance belongs to.  
itemid int //Id of the record tagged.  
ordering int //Tag ordering.  
flag int //Whether the tag is flagged as inappropriate.  
}  
}  
)  
)  
)contentsinfo Optional //Contents summary information.  
object {  
filescount int //Total number of files.  
filesize int //Total files size.  
lastmodified int //Last time files were modified.  
mimetypes //Files mime types.  
list of (  
string //File mime type.  
)repositorytype string Optional //The repository type for  
the main file.  
}  
}  
}  
})
```



XML-RPC (PHP structure)

```

Array
(
[0] =>
    Array
    (
        [id] => int
        [name] => string
        [visible] => int
        [summary] => string
        [summaryformat] => int
        [section] => int
        [hiddenbynumsections] => int
        [uservisible] => int
        [availabilityinfo] => string
        [modules] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [id] => int
                        [url] => string
                        [name] => string
                        [instance] => int
                        [contextid] => int
                        [description] => string
                        [visible] => int
                        [uservisible] => int
                        [availabilityinfo] => string
                        [visibleoncoursepage] => int
                        [modicon] => string
                        [modname] => string
                        [modplural] => string
                        [availability] => string
                        [indent] => int
                        [onclick] => string
                        [afterlink] => string
                        [customdata] => string
                        [noviewlink] => int
                        [completion] => int
                        [completiondata] =>
                            Array
                            (
                                [state] => int
                                [timecompleted] => int
                                [overrideby] => int
                                [valueused] => int
                                [hascompletion] => int
                                [isautomatic] => int
                                [istrackeduser] => int
                                [uservisible] => int
                                [details] =>
                                    Array
                                    (
                                        [0] =>
                                            Array
                                            (
                                                [rulename] => string
                                                [rulevalue] =>
                                                    Array
                                                    (
                                                        [status] => int
                                                        [description] => string
                                                    )
                                            )
                                    )
                            )
                        )
                    )
                )
            )
        [downloadcontent] => int
        [dates] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [label] => string
                        [timestamp] => int
                        [relativeto] => int
                        [dataid] => string
                    )
            )
        )
    )
)

```



```
[contents] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [type] => string
                [filename] => string
                [filepath] => string
                [filesize] => int
                [fileurl] => string
                [content] => string
                [timecreated] => int
                [timemodified] => int
                [sortorder] => int
                [mimetype] => string
                [isexternalfile] => int
                [repositorytype] => string
                [userid] => int
                [author] => string
                [license] => string
                [tags] =>
                  Array
                    (
                      [
                        [0] =>
                          Array
                            (
                              [
                                [id] => int
                                [name] => string
                                [rawname] => string
                                [isstandard] => int
                                [tagcollid] => int
                                [taginstanceid] => int
                                [taginstancecontextid] => int
                                [itemid] => int
                                [ordering] => int
                                [flag] => int
                              )
                            )
                      )
                    )
              )
            )
        [contentsinfo] =>
          Array
            (
              [
                [filecount] => int
                [filesize] => int
                [lastmodified] => int
                [mimetypes] =>
                  Array
                    (
                      [
                        [0] => string
                      )
                    )
                [repositorytype] => string
              )
            )
          )
        )
      )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="summary">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summaryformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="section">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hiddenbynumsections">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="uservisible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="availabilityinfo">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="modules">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="url">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="instance">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="contextid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="description">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="visible">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="uservisible">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="availabilityinfo">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="visibleoncoursepage">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="modicon">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="modname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="modplural">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="availability">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="indent">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="onclick">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```



```
</KEY>
<KEY name="afterlink">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customdata">
    <VALUE>string</VALUE>
</KEY>
<KEY name="noviewlink">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completion">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completiondata">
    <SINGLE>
        <KEY name="state">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecompleted">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="overrideby">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="valueused">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="hascompletion">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="isautomatic">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="istrackeduser">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="uservisible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="details">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="rulename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rulevalue">
                        <SINGLE>
                            <KEY name="status">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="description">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="downloadcontent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="dates">
    <MULTIPLE>
        <SINGLE>
            <KEY name="label">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timestamp">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="relativeto">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="dataid">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="contents">
    <MULTIPLE>
```



```

<SINGLE>
    <KEY name="type">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filename">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filepath">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filesize">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="fileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="content">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="timecreated">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="sortorder">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="mimetype">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="isexternalfile">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="repositorytype">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="userid">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="author">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="license">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="tags">
        <MULTIPLE>
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="rawname">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="isstandard">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="tagcollid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="taginstanceid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="taginstancecontextid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="ordering">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="flag">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>

```



```
</MULTIPLE>
</KEY>
<KEY name="contentsinfo">
<SINGLE>
    <KEY name="filecount">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="filesize">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="lastmodified">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="mimetypes">
        <MULTIPLE>
            <VALUE>string</VALUE>
        </MULTIPLE>
    </KEY>
    <KEY name="repositorytype">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_course_content_items

Fetch all the content items (activities, resources and their subtypes) for the activity picker

Arguments

courseid (Required)

ID of the course

General structure

XML-RPC (PHP structure)

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
content_items list of (
object {
id int //The id of the content item
name string //Name of the content item
title string //The string title of the content item, human readable
link string //The link to the content item creation page
icon string //Html containing the icon for the content item
help string //Html description / help for the content item
archetype string //The archetype of the module exposing the content item
componentname string //The name of the component exposing the content item
purpose string //The purpose of the component exposing the content item
favourite int //Has the user favourited the content item
legacyitem int //If this item was pulled from the old callback and has no item id.
recommended int //Has this item been recommended
}
)}
```

XML-RPC (PHP structure)

```
Array
(
[content_items] =>
Array
(
[0] =>
Array
(
[id] => int
[name] => string
[title] => string
[link] => string
[icon] => string
[help] => string
[archetype] => string
[componentname] => string
[purpose] => string
[favourite] => int
[legacyitem] => int
[recommended] => int
)
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="content_items">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="link">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="help">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="archetype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="componentname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="purpose">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="favourite">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="legacyitem">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="recommended">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_course_module

Return information about a course module

Arguments

cmid (Required)

The course module id

General structure

```
int //The course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

Response



General structure

```

object {
cm object {
id int //The course module id
course int //The course id
module int //The module type id
name string //The activity name
modname string //The module component name (forum, assign, etc..)
instance int //The activity instance id
section int //The module section id
sectionnum int //The module section number
groupmode int //Group mode
groupingid int //Grouping id
completion int //If completion is enabled
idnumber string Optional //Module id number
added int Optional //Time added
score int Optional //Score
indent int Optional //Indentation
visible int Optional //If visible
visibleoncoursepage int Optional //If visible on course page
visibleold int Optional //Visible old
completiongradeitemnumber int Optional //Completion grade item
completionpassgrade int Optional //Completion pass grade setting
completionview int Optional //Completion view setting
completionexpected int Optional //Completion time expected
showdescription int Optional //If the description is showed
downloadcontent int Optional //The download content value
availability string Optional //Availability settings
grade double Optional //Grade (max value or scale id)
scale string Optional //Scale items (if used)
gradepass string Optional //Grade to pass (float)
gradecat int Optional //Grade category
advancedgrading Optional //Advanced grading settings
list of (
object {
area string //Graddable area name
method string //Grading method
})
outcomes Optional //Outcomes information
list of (
object {
id string //Outcome id
name string //Outcome full name
scale string //Scale items
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
}

```



XML-RPC (PHP structure)

```

Array
(
    [cm] =>
        Array
            (
                [id] => int
                [course] => int
                [module] => int
                [name] => string
                [modname] => string
                [instance] => int
                [section] => int
                [sectionnum] => int
                [groupmode] => int
                [groupingid] => int
                [completion] => int
                [idnumber] => string
                [added] => int
                [score] => int
                [indent] => int
                [visible] => int
                [visibleoncoursepage] => int
                [visibleold] => int
                [completiongradeitemnumber] => int
                [completionpassgrade] => int
                [completionview] => int
                [completionexpected] => int
                [showdescription] => int
                [downloadcontent] => int
                [availability] => string
                [grade] => double
                [scale] => string
                [gradepass] => string
                [gradecat] => int
                [advancedgrading] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [area] => string
                                        [method] => string
                                    )
                        )
                )
            )
        [outcomes] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [id] => string
                                [name] => string
                                [scale] => string
                            )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="cm">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="course">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="module">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="modname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="instance">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="section">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="sectionnum">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupmode">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupingid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="completion">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="idnumber">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="added">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="score">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="indent">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visibleoncoursepage">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="visibleold">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="completiongradeitemnumber">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="completionpassgrade">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="completionview">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="completionexpected">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="showdescription">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="downloadcontent">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="availability">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="grade">
```



```

        <VALUE>double</VALUE>
    </KEY>
    <KEY name="scale">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="gradepass">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="gradecat">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="advancedgrading">
        <MULTIPLE>
            <SINGLE>
                <KEY name="area">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="method">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="outcomes">
        <MULTIPLE>
            <SINGLE>
                <KEY name="id">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="scale">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_course_module_by_instance

Return information about a given module name and instance id

Arguments**module** (Required)

The module name

General structure

```
string //The module name
```

XML-RPC (PHP structure)

```
[module] => string
```

REST (POST parameters)

```
module= string
```

instance (Required)

The module instance id

General structure

```
int //The module instance id
```

XML-RPC (PHP structure)

```
[instance] => int
```

REST (POST parameters)

```
instance= int
```

**Response**

General structure

```

object {
cm object {
id int //The course module id
course int //The course id
module int //The module type id
name string //The activity name
modname string //The module component name (forum, assign, etc..)
instance int //The activity instance id
section int //The module section id
sectionnum int //The module section number
groupmode int //Group mode
groupingid int //Grouping id
completion int //If completion is enabled
idnumber string Optional //Module id number
added int Optional //Time added
score int Optional //Score
indent int Optional //Indentation
visible int Optional //If visible
visibleoncoursepage int Optional //If visible on course page
visibleold int Optional //Visible old
completiongradeitemnumber int Optional //Completion grade item
completionpassgrade int Optional //Completion pass grade setting
completionview int Optional //Completion view setting
completionexpected int Optional //Completion time expected
showdescription int Optional //If the description is showed
downloadcontent int Optional //The download content value
availability string Optional //Availability settings
grade double Optional //Grade (max value or scale id)
scale string Optional //Scale items (if used)
gradepass string Optional //Grade to pass (float)
gradecat int Optional //Grade category
advancedgrading Optional //Advanced grading settings
list of (
object {
area string //Graddable area name
method string //Grading method
})
outcomes Optional //Outcomes information
list of (
object {
id string //Outcome id
name string //Outcome full name
scale string //Scale items
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
}

```



XML-RPC (PHP structure)

```

Array
(
    [cm] =>
        Array
            (
                [id] => int
                [course] => int
                [module] => int
                [name] => string
                [modname] => string
                [instance] => int
                [section] => int
                [sectionnum] => int
                [groupmode] => int
                [groupingid] => int
                [completion] => int
                [idnumber] => string
                [added] => int
                [score] => int
                [indent] => int
                [visible] => int
                [visibleoncoursepage] => int
                [visibleold] => int
                [completiongradeitemnumber] => int
                [completionpassgrade] => int
                [completionview] => int
                [completionexpected] => int
                [showdescription] => int
                [downloadcontent] => int
                [availability] => string
                [grade] => double
                [scale] => string
                [gradepass] => string
                [gradecat] => int
                [advancedgrading] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [area] => string
                                        [method] => string
                                    )
                        )
                )
            )
        [outcomes] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [id] => string
                                [name] => string
                                [scale] => string
                            )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="cm">
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="course">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="module">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="modname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="instance">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="section">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="sectionnum">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="groupmode">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="groupingid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completion">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="added">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="score">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="indent">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visibleoncoursepage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visibleold">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completiongradeitemnumber">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionpassgrade">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionview">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionexpected">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showdescription">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="downloadcontent">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="availability">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="grade">
```



```

        <VALUE>double</VALUE>
    </KEY>
    <KEY name="scale">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="gradepass">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="gradecat">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="advancedgrading">
        <MULTIPLE>
            <SINGLE>
                <KEY name="area">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="method">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="outcomes">
        <MULTIPLE>
            <SINGLE>
                <KEY name="id">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="scale">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_courses

Return course details

Arguments

options (Default to "Array ()")

options - operator OR is used

General structure

```
Default to "Array
(
)
" //options - operator OR is used
object {
ids  Optional //List of course id. If empty return all courses
      except front page course.
list of (
int   //Course id
)}
```

XML-RPC (PHP structure)

```
[options] =>
  Array
    (
      [ids] =>
        Array
          (
            [0] => int
          )
    )
)
```

REST (POST parameters)

options[ids][0]= int

Response

General structure

```

list of (
  //course
object {
id int    //course id
shortname string  //course short name
categoryid int   //category id
categorysortorder int  Optional //sort order into the category
fullname string   //full name
displayname string //course display name
idnumber string  Optional //id number
summary string   //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
format string   //course format: weeks, topics, social, site,..
showgrades int  Optional //1 if grades are shown, otherwise 0
newsitems int   Optional //number of recent items appearing on the course page
startdate int   //timestamp when the course start
enddate int    //timestamp when the course end
numsections int Optional //deprecated, use courseformatoptions) number of weeks/topics
maxbytes int   Optional //largest size of file that can be uploaded into the course
showreports int Optional //are activity report shown (yes = 1, no =0)
visible int    Optional //1: available to student, 0:not available
hiddensections int Optional //deprecated, use courseformatoptions) How the hidden sections in the course are displayed to students
groupmode int   Optional //no group, separate, visible
groupmodeforce int Optional //1: yes, 0: no
defaultgroupingid int Optional //default grouping id
timecreated int //timestamp when the course have been created
timemodified int //timestamp when the course have been modified
enablecompletion int Optional //Enabled, control via completion and activity settings. Disbaled,
                                not shown in activity settings.
completionnotify int Optional //1: yes 0: no
lang string    Optional //forced course Language
forcetheme string Optional //name of the force theme
courseformatoptions Optional //additional options for particular course format
list of (
object {
name string  //course format option name
value string  //course format option value
}
)showactivitydates int //Whether the activity dates are shown or not
showcompletionconditions int //Whether the activity completion conditions are shown or not
customfields Optional //Custom fields and associated values
list of (
object {
name string  //The name of the custom field
shortname string //The shortname of the custom field
type string   //The type of the custom field - text, checkbox...
valueraw string //The raw value of the custom field
value string  //The value of the custom field
}
)
)
)

```



XML-RPC (PHP structure)



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="categoryid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="categorysortorder">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="displayname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summary">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summaryformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="format">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="showgrades">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="newsitems">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="startdate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enddate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="numsections">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="maxbytes">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showreports">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hiddensections">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="groupmode">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="groupmodeforce">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="defaultgroupingid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enablecompletion">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionnotify">
        <VALUE>int</VALUE>
      </KEY>
```



```

</KEY>
<KEY name="lang">
    <VALUE>string</VALUE>
</KEY>
<KEY name="forcetheme">
    <VALUE>string</VALUE>
</KEY>
<KEY name="courseformatoptions">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="showactivitydates">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="valueraw">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message



REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_courses_by_field

Get courses matching a specific field (id/s, shortname, idnumber, category)

Arguments

field (Default to "")

The field to search can be left empty for all courses or: id: course id
ids: comma separated course ids
shortname: course short name
idnumber: course id number
category: category id the course belongs to

General structure

```
string Default to "" //The field to search can be left empty for all courses or:  
    id: course id  
    ids: comma separated course ids  
    shortname: course short name  
    idnumber: course id number  
    category: category id the course belongs to
```

XML-RPC (PHP structure)

```
[field] => string
```

REST (POST parameters)

```
field= string
```

value (Default to "")

The value to match

General structure

```
string Default to "" //The value to match
```

XML-RPC (PHP structure)

```
[value] => string
```

REST (POST parameters)

```
value= string
```

Response



```

General structure

object {
courses //Course
list of (
object {
id int //course id
fullname string //course full name
displayname string //course display name
shortname string //course short name
courseimage string Optional //Course image
categoryid int //category id
categoryname string //category name
sortorder int Optional //Sort order in the category
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
summaryfiles Optional //summary files in the summary field
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)overviewfiles //additional overview files attached to this course
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)showactivitydates int //Whether the activity dates are shown or not
showcompletionconditions int //Whether the activity completion conditions are shown or not
contacts //contact users
list of (
object {
id int //contact user id
fullname string //contact user fullname
}
)enrollmentmethods //enrollment methods List
list of (
string //enrollment method
)customfields Optional //Custom fields
list of (
object {
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
type string //The type of the custom field - text field, checkbox...
valueraw string //The raw value of the custom field
value string //The value of the custom field
}
)idnumber string Optional //Id number
format string Optional //Course format: weeks, topics, social, site,..
showgrades int Optional //1 if grades are shown, otherwise 0
newsitems int Optional //Number of recent items appearing on the course page
startdate int Optional //Timestamp when the course start
enddate int Optional //Timestamp when the course end
maxbytes int Optional //Largest size of file that can be uploaded into
showreports int Optional //Are activity report shown (yes = 1, no =0)
visible int Optional //1: available to student, 0:not available
groupmode int Optional //no group, separate, visible
groupmodeforce int Optional //1: yes, 0: no
defaultgroupingid int Optional //default grouping id
enablecompletion int Optional //Completion enabled? 1: yes 0: no
completionnotify int Optional //1: yes 0: no
lang string Optional //Forced course Language
theme string Optional //Name of the forced theme
marker int Optional //Current course marker
legacyfiles int Optional //If Legacy files are enabled
calendartype string Optional //Calendar type
timecreated int Optional //Time when the course was created
timemodified int Optional //Last time the course was updated

```

```
requested int Optional //If is a requested course
cacherev int Optional //Cache revision number
filters Optional //Course filters
list of (
object {
filter string //Filter plugin name
localstate int //Filter state: 1 for on, -1 for off, 0 if inherit
inheritedstate int //1 or 0 to use when Localstate is set to inherit
})
courseformatoptions Optional //Additional options for particular course format.
list of (
object {
name string //Course format option name.
value string //Course format option value.
})
))
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
))
```



XML-RPC (PHP structure)

```

Array
(
    [courses] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [fullname] => string
                            [displayname] => string
                            [shortname] => string
                            [courseimage] => string
                            [categoryid] => int
                            [categoryname] => string
                            [sortorder] => int
                            [summary] => string
                            [summaryformat] => int
                            [summaryfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [overviewfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [showactivitydates] => int
                            [showcompletionconditions] => int
                            [contacts] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [fullname] => string
                                                )
                                    )
                            [enrollmentmethods] =>
                                Array
                                    (
                                        [0] => string
                                    )
                            [customfields] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [name] => string
                                                    [shortname] => string
                                                    [type] => string
                                                    [valueraw] => string
                                                    [value] => string
                                                )
                                    )
                        )
                )
)

```



```
[idnumber] => string
[format] => string
[showgrades] => int
[newitems] => int
[startdate] => int
[enddate] => int
[maxbytes] => int
[showreports] => int
[visible] => int
[groupmode] => int
[groupmodeforce] => int
[defaultgroupingid] => int
[enablecompletion] => int
[completionnotify] => int
[lang] => string
[theme] => string
[marker] => int
[legacyfiles] => int
[calendartype] => string
[timecreated] => int
[timemodified] => int
[requested] => int
[cacherev] => int
[filters] =>
  Array
    (
      [0] =>
        Array
          (
            [filter] => string
            [localstate] => int
            [inheritedstate] => int
          )
    )
  [courseformatoptions] =>
    Array
      (
        [0] =>
          Array
            (
              [name] => string
              [value] => string
            )
      )
    )
  )
)
[warnings] =>
  Array
    (
      [0] =>
        Array
          (
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
          )
    )
  )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="courses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="displayname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="shortname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="courseimage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="categoryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="categoryname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="summary">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="summaryformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="summaryfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="repositorytype">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                    <KEY name="overviewfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="showactivitydates">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contacts">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="enrollmentmethods">
    <MULTIPLE>
        <VALUE>string</VALUE>
    </MULTIPLE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="valueraw">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="idnumber">
    <VALUE>string</VALUE>
</KEY>
<KEY name="format">
    <VALUE>string</VALUE>
</KEY>
<KEY name="showgrades">
    <VALUE>int</VALUE>
</KEY>
<KEY name="newsitems">
    <VALUE>int</VALUE>
</KEY>
<KEY name="startdate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="enddate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxbytes">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showreports">
    <VALUE>int</VALUE>
</KEY>
```



```
<KEY name="visible">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupmode">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupmodeforce">
    <VALUE>int</VALUE>
</KEY>
<KEY name="defaultgroupingid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="enablecompletion">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completionnotify">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lang">
    <VALUE>string</VALUE>
</KEY>
<KEY name="theme">
    <VALUE>string</VALUE>
</KEY>
<KEY name="marker">
    <VALUE>int</VALUE>
</KEY>
<KEY name="legacyfiles">
    <VALUE>int</VALUE>
</KEY>
<KEY name="calendartype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requested">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cacherev">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filters">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filter">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="localstate">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="inheritedstate">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="courseformatoptions">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

        </KEY>
        <KEY name="warningcode">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="message">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_enrolled_courses_by_timeline_classification

List of enrolled courses for the given timeline classification (past, inprogress, or future).

Arguments

classification (Required)

future, inprogress, or past

General structure

```
string //future, inprogress, or past
```



XML-RPC (PHP structure)

```
[classification] => string
```

REST (POST parameters)

```
classification= string
```

limit (Default to "0")

Result set limit

General structure

```
int Default to "0" //Result set limit
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

offset (Default to "0")

Result set offset

General structure

```
int Default to "0" //Result set offset
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

sort (Default to "null")

Sort string

General structure

```
string Default to "null" //Sort string
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

customfieldname (Default to "null")

Used when classification = customfield

General structure

```
string Default to "null" //Used when classification = customfield
```

XML-RPC (PHP structure)

```
[customfieldname] => string
```

REST (POST parameters)

```
customfieldname= string
```

customfieldvalue (Default to "null")

Used when classification = customfield

General structure

```
string Default to "null" //Used when classification = customfield
```

XML-RPC (PHP structure)

```
[customfieldvalue] => string
```

REST (POST parameters)

```
customfieldvalue= string
```

searchvalue (Default to "null")

The value a user wishes to search against

General structure

```
string Default to "null" //The value a user wishes to search against
```

XML-RPC (PHP structure)

```
[searchvalue] => string
```

REST (POST parameters)

```
searchvalue= string
```

Response**General structure**

```
object {
courses //Course
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
)nextoffset int //Offset for the next request
}
```



XML-RPC (PHP structure)

```
Array
(
    [courses] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [fullname] => string
                            [shortname] => string
                            [idnumber] => string
                            [summary] => string
                            [summaryformat] => int
                            [startdate] => int
                            [enddate] => int
                            [visible] => int
                            [showactivitydates] => int
                            [showcompletionconditions] => int
                            [pdfexportfont] => string
                            [fullnamedisplay] => string
                            [viewurl] => string
                            [courseimage] => string
                            [progress] => int
                            [hasprogress] => int
                            [isfavourite] => int
                            [hidden] => int
                            [timeaccess] => int
                            [showshortname] => int
                            [coursecategory] => string
                        )
                )
            )
        [nextoffset] => int
    )
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="courses">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="summary">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="summaryformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="startdate">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="enddate">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="visible">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showactivitydates">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showcompletionconditions">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="pdlexportfont">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="fullnamedisplay">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="viewurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="courseimage">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="progress">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="hasprogress">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isfavourite">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="hidden">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeaccess">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showshortname">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="coursecategory">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="nextoffset">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
```



</RESPONSE>

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_enrolled_courses_with_action_events_by_timeline_classification

List of enrolled courses with action events in a given timeframe, for the given timeline classification.

Arguments

classification (Required)

future, inprogress, or past

General structure

```
string //future, inprogress, or past
```

XML-RPC (PHP structure)

```
[classification] => string
```

REST (POST parameters)

```
classification= string
```

**limit** (Default to "0")

Result set limit

General structure

```
int Default to "0" //Result set limit
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

offset (Default to "0")

Result set offset

General structure

```
int Default to "0" //Result set offset
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

sort (Default to "null")

Sort string

General structure

```
string Default to "null" //Sort string
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

customfieldname (Default to "null")

Used when classification = customfield

General structure

```
string Default to "null" //Used when classification = customfield
```

XML-RPC (PHP structure)

```
[customfieldname] => string
```

REST (POST parameters)

```
customfieldname= string
```



customfieldvalue (Default to "null")

Used when classification = customfield

General structure

```
string Default to "null" //Used when classification = customfield
```

XML-RPC (PHP structure)

```
[customfieldvalue] => string
```

REST (POST parameters)

```
customfieldvalue= string
```

searchvalue (Default to "null")

The value a user wishes to search against

General structure

```
string Default to "null" //The value a user wishes to search against
```

XML-RPC (PHP structure)

```
[searchvalue] => string
```

REST (POST parameters)

```
searchvalue= string
```

eventsfrom (Default to "null")

Optional starting timestamp for action events

General structure

```
int Default to "null" //Optional starting timestamp for action events
```

XML-RPC (PHP structure)

```
[eventsfrom] => int
```

REST (POST parameters)

```
eventsfrom= int
```

eventsto (Default to "null")

Optional ending timestamp for action events

General structure

```
int Default to "null" //Optional ending timestamp for action events
```

XML-RPC (PHP structure)

```
[eventsto] => int
```

REST (POST parameters)

```
eventsto= int
```

**Response**

General structure

```
object {
courses //Course
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
)nextoffset int //Offset for the next request
morecoursesavailable int //Whether more courses with events exist within the provided parameters
}
```

XML-RPC (PHP structure)

```

Array
(
    [courses] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [id] => int
                        [fullname] => string
                        [shortname] => string
                        [idnumber] => string
                        [summary] => string
                        [summaryformat] => int
                        [startdate] => int
                        [enddate] => int
                        [visible] => int
                        [showactivitydates] => int
                        [showcompletionconditions] => int
                        [pdlexportfont] => string
                        [fullnamedisplay] => string
                        [viewurl] => string
                        [courseimage] => string
                        [progress] => int
                        [hasprogress] => int
                        [isfavourite] => int
                        [hidden] => int
                        [timeaccess] => int
                        [showshortname] => int
                        [coursecategory] => string
                    )
            )
        )
    )
)
[nextoffset] => int
[morecoursesavailable] => int
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="courses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="shortname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="summary">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="summaryformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="startdate">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="enddate">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="visible">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="showactivitydates">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="showcompletionconditions">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="pdlexportfont">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullnamedisplay">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="viewurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="courseimage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="progress">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="hasprogress">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isfavourite">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="hidden">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeaccess">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="showshortname">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursecategory">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="nextoffset">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="morecoursesavailable">
            <VALUE>int</VALUE>
        </KEY>
```



```
</SINGLE>
</RESPONSE>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_course_get_enrolled_users_by_cmid`

List users by course module id, filter by group and active enrolment status.

Arguments`cmid` (Required)

id of the course module

General structure

```
int //id of the course module
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

`groupid` (Default to "0")

id of the group

General structure

```
int Default to "0" //id of the group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

`onlyactive` (Default to "")

whether to return only active users or all.

General structure

```
int Default to "" //whether to return only active users or all.
```

XML-RPC (PHP structure)

```
[onlyactive] => int
```

REST (POST parameters)

```
onlyactive= int
```

Response**General structure**

```
object {
    users list of (
        object {
            id int //ID of the user
            profileimage string Optional //The location of the users larger image
            fullname string Optional //The full name of the user
            firstname string Optional //The first name(s) of the user
            lastname string Optional //The family name of the user
        }
    )warnings Optional //list of warnings
    list of (
        //warning
    )object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [users] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [profileimage] => string
                        [fullname] => string
                        [firstname] => string
                        [lastname] => string
                    )
                )
            )
        [warnings] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                )
            )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="profileimage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="firstname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="lastname">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_module

Returns html with one activity module on course page

Arguments

id (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

sectionreturn (Default to "null")

section to return to

General structure

```
int Default to "null" //section to return to
```

XML-RPC (PHP structure)

```
[sectionreturn] => int
```

REST (POST parameters)

```
sectionreturn= int
```

Response

html to replace the current module with

General structure

```
string //html to replace the current module with
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_recent_courses

List of courses a user has accessed most recently.

Arguments**userid** (Default to "0")

id of the user, default to current user

General structure

```
int Default to "0" //id of the user, default to current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

limit (Default to "0")

result set limit

General structure

```
int Default to "0" //result set limit
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

offset (Default to "0")

Result set offset

General structure

```
int Default to "0" //Result set offset
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

sort (Default to "null")

Sort string

General structure

```
string Default to "null" //Sort string
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

Response

Courses

General structure

```
//Courses
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
})
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [fullname] => string
                [shortname] => string
                [idnumber] => string
                [summary] => string
                [summaryformat] => int
                [startdate] => int
                [enddate] => int
                [visible] => int
                [showactivitydates] => int
                [showcompletionconditions] => int
                [pdfexportfont] => string
                [fullnamedisplay] => string
                [viewurl] => string
                [courseimage] => string
                [progress] => int
                [hasprogress] => int
                [isfavourite] => int
                [hidden] => int
                [timeaccess] => int
                [showshortname] => int
                [coursecategory] => string
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summary">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summaryformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="startdate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enddate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showactivitydates">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showcompletionconditions">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="pdfexportfont">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullnamedisplay">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="viewurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="courseimage">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="progress">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hasprogress">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isfavourite">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hidden">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timeaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showshortname">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="coursecategory">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_updates_since

Check if there are updates affecting the user for the given course since the given time stamp.

Arguments**courseid** (Required)

Course id to check

General structure

```
int //Course id to check
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

since (Required)

Check updates since this time stamp

General structure

```
int //Check updates since this time stamp
```

**XML-RPC (PHP structure)**

```
[since] => int
```

REST (POST parameters)

```
since= int
```

filter (Default to "Array ()")

Check only for updates in these areas

```
General structure

Default to "Array
(
)
" //Check only for updates in these areas
list of (
string //Area name: configuration, fileareas, completion, ratings, comments,
gradeitems, outcomes
)
```

XML-RPC (PHP structure)

```
[filter] =>
    Array
    (
        [0] => string
    )
```

REST (POST parameters)

```
filter[0]= string
```

Response

```
General structure

object {
instances list of (
object {
contextlevel string //The context level
id int //Instance id
updates list of (
object {
name string //Name of the area updated.
timeupdated int Optional //Last time was updated
itemids Optional //The ids of the items updated
list of (
int //Instance id
)}
)
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [instances] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [contextlevel] => string
                                [id] => int
                                [updates] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [name] => string
                                                        [timeupdated] => int
                                                        [itemids] =>
                                                            Array
                                                            (
                                                                [
                                                                    [0] => int
                                                                )
                                                            )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="instances">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="contextlevel">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="updates">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timeupdated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="itemids">
                                    <MULTIPLE>
                                        <VALUE>int</VALUE>
                                    </MULTIPLE>
                                </KEY>
                                </SINGLE>
                            </MULTIPLE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
            <KEY name="warnings">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="item">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="itemid">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="warningcode">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="message">
                            <VALUE>string</VALUE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_get_user_administration_options

Return a list of administration options in a set of courses that are available or not for the current user.

Arguments

courseids (Required)

General structure

```
list of (
int //Course id.
)
```

XML-RPC (PHP structure)

```
[courseids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

courseids[0]= int

Response**General structure**

```
object {
courses //List of courses
list of (
object {
id int //Course id
options list of (
object {
name string //Option name
available int //Whether the option is available or not
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [courses] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [options] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [name] => string
                                                        [available] => int
                                                    )
                                                )
                                        )
                                    )
                )
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="courses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="options">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="available">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_get_user_navigation_options

Return a list of navigation options in a set of courses that are available or not for the current user.

Arguments

courseids (Required)

General structure

```
list of (
int //Course id.
)
```

XML-RPC (PHP structure)

```
[courseids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
courseids[0]= int
```

Response

General structure

```
object {
courses //List of courses
list of (
object {
id int //Course id
options list of (
object {
name string //Option name
available int //Whether the option is available or not
}
)
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}
```

XML-RPC (PHP structure)

```
Array
(
    [
        [courses] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [options] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [name] => string
                                                        [available] => int
                                                    )
                                                )
                                        )
                                    )
                )
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="courses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="options">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="available">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_import_course

Import course data from a course into another course. Does not include any user data.

Arguments**importfrom** (Required)

the id of the course we are importing from

General structure

```
int //the id of the course we are importing from
```

XML-RPC (PHP structure)

```
[importfrom] => int
```

REST (POST parameters)

```
importfrom= int
```

importto (Required)

the id of the course we are importing to

General structure

```
int //the id of the course we are importing to
```

XML-RPC (PHP structure)

```
[importto] => int
```

REST (POST parameters)

```
importto= int
```

deletecontent (Default to "0")

whether to delete the course content where we are importing to (default to 0 = No)

General structure

```
int Default to "0" //whether to delete the course content where we are importing to (default to 0 = No)
```

XML-RPC (PHP structure)

```
[deletecontent] => int
```

**REST (POST parameters)**

```
deletecontent= int
```

options (Default to "Array ()")

Course import options

```
General structure

Default to "Array
(
)
" //Course import options
list of (
object {
name string //The backup option name:
                    "activities" (int) Include course activites (default to 1 that is equal to yes),
                    "blocks" (int) Include course blocks (default to 1 that is equal to yes),
                    "filters" (int) Include course filters (default to 1 that is equal to yes)
value string //the value for the option 1 (yes) or 0 (no)
}
)
```

XML-RPC (PHP structure)

```
[options] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_remove_content_item_from_user_favourites

Removes a content item (activity, resource or their subtypes) from the favourites for the user.

Arguments**componentname** (Required)

frankenstyle name of the component to which the content item belongs

General structure

```
string //frankenstyle name of the component to which the content item belongs
```

XML-RPC (PHP structure)

```
[componentname] => string
```

REST (POST parameters)

```
componentname= string
```

contentitemid (Required)

id of the content item

General structure

```
int //id of the content item
```

XML-RPC (PHP structure)

```
[contentitemid] => int
```

REST (POST parameters)

```
contentitemid= int
```

Response**General structure**

```
object {
  id int //The id of the content item
  name string //Name of the content item
  title string //The string title of the content item, human readable
  link string //The link to the content item creation page
  icon string //Html containing the icon for the content item
  help string //Html description / help for the content item
  archetype string //The archetype of the module exposing the content item
  componentname string //The name of the component exposing the content item
  purpose string //The purpose of the component exposing the content item
  favourite int //Has the user favourited the content item
  legacyitem int //If this item was pulled from the old callback and has no item id.
  recommended int //Has this item been recommended
}
```

**XML-RPC (PHP structure)**

```
Array
(
    [id] => int
    [name] => string
    [title] => string
    [link] => string
    [icon] => string
    [help] => string
    [archetype] => string
    [componentname] => string
    [purpose] => string
    [favourite] => int
    [legacyitem] => int
    [recommended] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="title">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="link">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="icon">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="help">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="archetype">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="componentname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="purpose">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="favourite">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="legacyitem">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="recommended">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_course_search_courses

Search courses by (name, module, block, tag)

Arguments

criterianame (Required)

criteria name (search, modulelist (only admins), blocklist (only admins), tagid)

General structure

```
string //criteria name
          (search, moduleList (only admins), blockList (only admins), tagId)
```

XML-RPC (PHP structure)

```
[criterianame] => string
```

REST (POST parameters)

```
criterianame= string
```

criteriavalue (Required)

criteria value

General structure

```
string //criteria value
```

XML-RPC (PHP structure)

```
[criteriavalue] => string
```

REST (POST parameters)

```
criteriavalue= string
```

page (Default to "0")

page number (0 based)

General structure

```
int Default to "0" //page number (0 based)
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "0")

items per page

General structure

```
int Default to "0" //items per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

requiredcapabilities (Default to "Array ()")

Optional list of required capabilities (used to filter the list)

```
General structure

Default to "Array
(
)
" //Optional List of required capabilities (used to filter the list)
list of (
string //Capability string used to filter courses by permission
)
```

XML-RPC (PHP structure)

```
[requiredcapabilities] =>
Array
(
    [0] => string
)
```

REST (POST parameters)

```
requiredcapabilities[0]= string
```

limittoenrolled (Default to "0")

limit to enrolled courses

General structure

```
int Default to "0" //Limit to enrolled courses
```

XML-RPC (PHP structure)

```
[limittoenrolled] => int
```

REST (POST parameters)

```
limittoenrolled= int
```

onlywithcompletion (Default to "0")

limit to courses where completion is enabled

General structure

```
int Default to "0" //Limit to courses where completion is enabled
```



XML-RPC (PHP structure)

```
[onlywithcompletion] => int
```

REST (POST parameters)

```
onlywithcompletion= int
```

Response

```

General structure

object {
  total int //total course count
  courses //course
  list of (
    object {
      id int //course id
      fullname string //course full name
      displayname string //course display name
      shortname string //course short name
      courseimage string Optional //Course image
      categoryid int //category id
      categoryname string //category name
      sortorder int Optional //Sort order in the category
      summary string //summary
      summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
      summaryfiles Optional //summary files in the summary field
      list of (
        //File
      object {
        filename string Optional //File name.
        filepath string Optional //File path.
        filesize int Optional //File size.
        fileurl string Optional //Downloadable file url.
        timemodified int Optional //Time modified.
        mimetype string Optional //File mime type.
        isexternalfile int Optional //Whether is an external file.
        repositorytype string Optional //The repository type for external files.
      }
    )overviewfiles //additional overview files attached to this course
    list of (
      //File
    object {
      filename string Optional //File name.
      filepath string Optional //File path.
      filesize int Optional //File size.
      fileurl string Optional //Downloadable file url.
      timemodified int Optional //Time modified.
      mimetype string Optional //File mime type.
      isexternalfile int Optional //Whether is an external file.
      repositorytype string Optional //The repository type for external files.
    }
  )showactivitydates int //Whether the activity dates are shown or not
  showcompletionconditions int //Whether the activity completion conditions are shown or not
  contacts //contact users
  list of (
  object {
    id int //contact user id
    fullname string //contact user fullname
  }
  )enrollmentmethods //enrollment methods list
  list of (
    string //enrollment method
  )customfields Optional //Custom fields
  list of (
  object {
    name string //The name of the custom field
    shortname string //The shortname of the custom field - to be able to build the field class in the code
    type string //The type of the custom field - text field, checkbox...
    valueraw string //The raw value of the custom field
    value string //The value of the custom field
  }
  ))
  )warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
  ))
}

```



XML-RPC (PHP structure)

```

Array
(
    [total] => int
    [courses] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [displayname] => string
                    [shortname] => string
                    [courseimage] => string
                    [categoryid] => int
                    [categoryname] => string
                    [sortorder] => int
                    [summary] => string
                    [summaryformat] => int
                    [summaryfiles] =>
                        Array
                        (
                            [0] =>
                                Array
                                (
                                    [filename] => string
                                    [filepath] => string
                                    [filesize] => int
                                    [fileurl] => string
                                    [timemodified] => int
                                    [mimetype] => string
                                    [isexternalfile] => int
                                    [repositorytype] => string
                                )
                        )
                )
            [overviewfiles] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [filename] => string
                            [filepath] => string
                            [filesize] => int
                            [fileurl] => string
                            [timemodified] => int
                            [mimetype] => string
                            [isexternalfile] => int
                            [repositorytype] => string
                        )
                )
        )
    [showactivitydates] => int
    [showcompletionconditions] => int
    [contacts] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                )
        )
    [enrollmentmethods] =>
        Array
        (
            [0] => string
        )
    [customfields] =>
        Array
        (
            [0] =>
                Array
                (
                    [name] => string
                    [shortname] => string
                    [type] => string
                    [valueraw] => string
                    [value] => string
                )
        )
)

```



```
        )
    )
[warnings] =>
  Array
  (
    [0] =>
      Array
      (
        [
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
      )
  )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="total">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="courses">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="displayname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="shortname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="courseimage">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="categoryid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="categoryname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="summary">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="summaryformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="summaryfiles">
            <MULTIPLE>
              <SINGLE>
                <KEY name="filename">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filepath">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filesize">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="fileurl">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="timemodified">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="mimetype">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="isexternalfile">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="repositorytype">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
          <KEY name="overviewfiles">
            <MULTIPLE>
              <SINGLE>
                <KEY name="filename">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filepath">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filesize">
                  <VALUE>int</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="showactivitydates">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="contacts">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
<KEY name="enrollmentmethods">
    <MULTIPLE>
        <VALUE>string</VALUE>
    </MULTIPLE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="valueraw">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_set_favourite_courses

Add a list of courses to the list of favourite courses.

Arguments

courses (Required)

General structure

```
list of (
object {
id int //course ID
favourite int //favourite status
}
)
```

XML-RPC (PHP structure)

```
[courses] =>
Array
(
[0] =>
Array
(
[id] => int
[favourite] => int
)
)
```

**REST (POST parameters)**

```
courses[0][id]= int
courses[0][favourite]= int
```

Response

General structure

```
object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_toggle_activity_recommendation

Adds or removes an activity as a recommendation in the activity chooser.

Arguments**area** (Required)

The favourite area (itemtype)

General structure

```
string //The favourite area (itemtype)
```

XML-RPC (PHP structure)

```
[area] => string
```

REST (POST parameters)

```
area= string
```

id (Required)

id of the activity or whatever

General structure

```
int //id of the activity or whatever
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

**Response****General structure**

```
object {
  id int //id of the activity or whatever
  area string //The favourite area (itemtype)
  status int //If created or deleted
}
```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [area] => string
    [status] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="area">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_course_update_categories**Update categories****Arguments**

categories (Required)

**General structure**

```
list of (
object {
id int //course id
name string Optional //category name
idnumber string Optional //category id number
parent int Optional //parent category id
description string Optional //category description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
theme string Optional //the category theme. This option must be enabled on moodle
}
)
```

XML-RPC (PHP structure)

```
[categories] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [name] => string
                        [idnumber] => string
                        [parent] => int
                        [description] => string
                        [descriptionformat] => int
                        [theme] => string
                    )
        )
    )
```

REST (POST parameters)

```
categories[0][id]= int
categories[0][name]= string
categories[0][idnumber]= string
categories[0][parent]= int
categories[0][description]= string
categories[0][descriptionformat]= int
categories[0][theme]= string
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

core_course_update_courses

Update courses

Arguments**courses** (Required)

courses to update

General structure

```
//courses to update
list of (
object {
id int //ID of the course
fullname string Optional //full name
shortname string Optional //course short name
categoryid int Optional //category id
idnumber string Optional //id number
summary string Optional //summary
summaryformat int Optional //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
format string Optional //course format: weeks, topics, social, site,..
showgrades int Optional //1 if grades are shown, otherwise 0
newsitems int Optional //number of recent items appearing on the course page
startdate int Optional //timestamp when the course start
enddate int Optional //timestamp when the course end
numsections int Optional //(deprecated, use courseformatoptions) number of weeks/topics
maxbytes int Optional //largest size of file that can be uploaded into the course
showreports int Optional //are activity report shown (yes = 1, no =0)
visible int Optional //1: available to student, 0:not available
hiddensections int Optional //(deprecated, use courseformatoptions) How the hidden sections in the course are
displayed to students
groupmode int Optional //no group, separate, visible
groupmodeforce int Optional //1: yes, 0: no
defaultgroupingid int Optional //default grouping id
enablecompletion int Optional //Enabled, control via completion and activity settings. Disabled,
not shown in activity settings.
completionnotify int Optional //1: yes 0: no
lang string Optional //forced course language
forcetheme string Optional //name of the force theme
courseformatoptions Optional //additional options for particular course format
list of (
object {
name string //course format option name
value string //course format option value
})
customfields Optional //Custom fields
list of (
object {
shortname string //The shortname of the custom field
value string //The value of the custom field
})
)}
```



XML-RPC (PHP structure)

```
[courses] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [fullname] => string
                        [shortname] => string
                        [categoryid] => int
                        [idnumber] => string
                        [summary] => string
                        [summaryformat] => int
                        [format] => string
                        [showgrades] => int
                        [newitems] => int
                        [startdate] => int
                        [enddate] => int
                        [numsections] => int
                        [maxbytes] => int
                        [showreports] => int
                        [visible] => int
                        [hiddensections] => int
                        [groupmode] => int
                        [groupmodeforce] => int
                        [defaultgroupingid] => int
                        [enablecompletion] => int
                        [completionnotify] => int
                        [lang] => string
                        [forcetheme] => string
                        [courseformatoptions] =>
                            Array
                                (
                                    [0] =>
                                        Array
                                            (
                                                [name] => string
                                                [value] => string
                                            )
                                )
            )
        )
    [customfields] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [shortname] => string
                            [value] => string
                        )
            )
        )
    )
)
```

```
REST (POST parameters)

courses[0][id]= int
courses[0][fullname]= string
courses[0][shortname]= string
courses[0][categoryid]= int
courses[0][idnumber]= string
courses[0][summary]= string
courses[0][summaryformat]= int
courses[0][format]= string
courses[0][showgrades]= int
courses[0][newsitems]= int
courses[0][startdate]= int
courses[0][enddate]= int
courses[0][numsections]= int
courses[0][maxbytes]= int
courses[0][showreports]= int
courses[0][visible]= int
courses[0][hiddensections]= int
courses[0][groupmode]= int
courses[0][groupmodeforce]= int
courses[0][defaultgroupingid]= int
courses[0][enablecompletion]= int
courses[0][completionnotify]= int
courses[0][lang]= string
courses[0][forcetheme]= string
courses[0][courseformatoptions][0][name]= string
courses[0][courseformatoptions][0][value]= string
courses[0][customfields][0][shortname]= string
courses[0][customfields][0][value]= string
```

Response

General structure

```
object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_course_view_course

Log that the course was viewed

Arguments

courseid (Required)

id of the course

General structure

int //id of the course

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

```
courseid= int
```

sectionnumber (Default to "0")

section number

General structure

```
int Default to "0" //section number
```

XML-RPC (PHP structure)

```
[sectionnumber] => int
```

REST (POST parameters)

```
sectionnumber= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**core_courseformat_file_handlers**

Get the current course file hanlders.

Arguments**courseid** (Required)

course id

General structure

int //course id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
list of (
object {
extension string //File extension
module string //Target module
message string //Output message
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[extension] => string
[module] => string
[message] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="extension">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="module">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_courseformat_get_state

Get the current course state.

Arguments

courseid (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

Encoded course state JSON

General structure

```
string //Encoded course state JSON
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_courseformat_update_course

Update course contents.

Arguments

action (Required)

action: cm_hide, cm_show, section_hide, section_show, cm_moveleft...

General structure

```
string //action: cm_hide, cm_show, section_hide, section_show, cm_moveleft...
```

XML-RPC (PHP structure)

```
[action] => string
```

REST (POST parameters)

```
action= string
```

courseid (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

ids (Default to "Array ()")

Affected ids

General structure

```
Default to "Array
(
)
" //Affected ids
list of (
int //Target id
)
```

**XML-RPC (PHP structure)**

```
[ids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
ids[0]= int
```

targetsectionid (Default to "null")

Optional target section id

General structure

```
int Default to "null" //Optional target section id
```

XML-RPC (PHP structure)

```
[targetsectionid] => int
```

REST (POST parameters)

```
targetsectionid= int
```

targetcmid (Default to "null")

Optional target cm id

General structure

```
int Default to "null" //Optional target cm id
```

XML-RPC (PHP structure)

```
[targetcmid] => int
```

REST (POST parameters)

```
targetcmid= int
```

Response

Encoded course update JSON

General structure

```
string //Encoded course update JSON
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_create_userfeedback_action_record

Record the action that the user takes in the user feedback notification for future use.

Arguments**action** (Required)

The action taken by user

General structure

```
string //The action taken by user
```

XML-RPC (PHP structure)

```
[action] => string
```

REST (POST parameters)

```
action= string
```

contextid (Required)

The context id of the page the user is in

General structure

```
int //The context id of the page the user is in
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

**Response****Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_create_category

Creates a new category

Arguments

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

area (Required)

area

General structure

```
string //area
```

XML-RPC (PHP structure)

```
[area] => string
```

REST (POST parameters)

```
area= string
```



itemid (Required)

itemid

General structure

```
int //itemid
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

Response

Id of the category

General structure

```
int //Id of the category
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_delete_category

Deletes a category



Arguments

id (Required)

category ID to delete

General structure

```
int //category ID to delete
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_delete_field

Deletes an entry

Arguments

id (Required)

Custom field ID to delete

General structure

```
int //Custom field ID to delete
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```



Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_move_category

Drag and drop categories

Arguments**id** (Required)

Category ID to move

General structure

```
int //Category ID to move
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

beforeid (Default to "0")

Id of the category before which it needs to be moved

General structure

```
int Default to "0" //Id of the category before which it needs to be moved
```

XML-RPC (PHP structure)

```
[beforeid] => int
```

REST (POST parameters)

```
beforeid= int
```

**Response****Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_move_field

Drag and drop

Arguments

id (Required)

Id of the field to move

General structure

```
int //Id of the field to move
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

categoryid (Required)

New parent category id

General structure

```
int //New parent category id
```

XML-RPC (PHP structure)

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

beforeid (Default to "0")

Id of the field before which it needs to be moved

General structure

```
int Default to "0" //Id of the field before which it needs to be moved
```

XML-RPC (PHP structure)

```
[beforeid] => int
```

REST (POST parameters)

```
beforeid= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_customfield_reload_template

Reloads template

Arguments

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

area (Required)

area

General structure

```
string //area
```

XML-RPC (PHP structure)

```
[area] => string
```

REST (POST parameters)

```
area= string
```

itemid (Required)

itemid

General structure

```
int //itemid
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

Response**General structure**

```
object {
  component string //component
  area string //area
  itemid int //itemid
  usescategories int //view has categories
  categories list of (
    object {
      id int //id
      nameeditable string //inplace editable name
      addfieldmenu string //addfieldmenu
      fields list of (
        object {
          name string //name
          shortname string //shortname
          type string //type
          id int //id
        }
      )
    })
}
```

XML-RPC (PHP structure)

```
Array
(
  [component] => string
  [area] => string
  [itemid] => int
  [usescategories] => int
  [categories] =>
    Array
      (
        [0] =>
          Array
            (
              [id] => int
              [nameeditable] => string
              [addfieldmenu] => string
              [fields] =>
                Array
                  (
                    [0] =>
                      Array
                        (
                          [name] => string
                          [shortname] => string
                          [type] => string
                          [id] => int
                        )
                  )
            )
      )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="area">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="usescategories">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="categories">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="nameeditable">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="addfieldmenu">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fields">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="shortname">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="type">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_dynamic_tabs_get_content

Returns the content for a dynamic tab

Arguments**tab** (Required)

Tab class

General structure

```
string //Tab class
```

XML-RPC (PHP structure)

```
[tab] => string
```

REST (POST parameters)

```
tab= string
```

jsondata (Required)

Json-encoded data

General structure

```
string //Json-encoded data
```

XML-RPC (PHP structure)

```
[jsondata] => string
```

REST (POST parameters)

```
jsondata= string
```

Response**General structure**

```
object {
  template string //Template name
  content string //JSON-encoded data for template
  javascript string //JavaScript fragment
}
```

**XML-RPC (PHP structure)**

```
Array
(
  [template] => string
  [content] => string
  [javascript] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="template">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="content">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="javascript">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_enrol_get_course_enrolment_methods

Get the list of course enrolment methods

Arguments

courseid (Required)

Course id

**General structure**

int //Course id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

Response

General structure

```
list of (
object {
id int //id of course enrolment instance
courseid int //id of course
type string //type of enrolment plugin
name string //name of enrolment plugin
status string //status of enrolment plugin
wsfunction string Optional //webservice function to get more information
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
    (
        [
            [id] => int
            [courseid] => int
            [type] => string
            [name] => string
            [status] => string
            [wsfunction] => string
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="status">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="wsfunction">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_enrol_get_enrolled_users

Get enrolled users by course id.

Arguments**courseid** (Required)

course id

General structure

int //course id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

options (Default to "Array ()")

Option names: * withcapability (string) return only users with this capability. This option requires 'moodle/role:review' on the course context. * groupid (integer) return only users in this group id. If the course has groups enabled and this param isn't defined, returns all the viewable users. This option requires 'moodle/site:accessallgroups' on the course context if the user doesn't belong to the group. * onlyactive (integer) return only users with active enrolments and matching time restrictions. This option requires 'moodle/course:enrolreview' on the course context. Please note that this option can't be used together with onlysuspended (only one can be active). * onlysuspended (integer) return only suspended users. This option requires 'moodle/course:enrolreview' on the course context. Please note that this option can't be used together with onlyactive (only one can be active). * userfields ('string, string, ...') return only the values of these user fields. * limitfrom (integer) sql limit from. * limitnumber (integer) maximum number of returned users. * sortby (string) sort by id, firstname or lastname. For ordering like the site does, use siteorder. * sortdirection (string) ASC or DESC

General structure

Default to "Array"

(

)

" //Option names:

* withcapability (string) return only users with this capability. This option requires 'moodle/role:review' on the course context.
 * groupid (integer) return only users in this group id. If the course has group led and this param isn't defined, returns all the viewable users.
 This option requires 'moodle/site:accessallgroups' on the course context if the user doesn't belong to the group.
 * onlyactive (integer) return only users with active enrolments and matching time restrictions.
 This option requires 'moodle/course:enrolreview' on the course context.
 Please note that this option can't be used together with onlysuspended (only one can be active).
 * onlysuspended (integer) return only suspended users. This option requires 'moodle/course:enrolreview' on the course context. Please note that this option can't be used together with onlyactive (only one can be active).
 * userfields ('string, string, ...') return only the values of these user fields.
 * Limitfrom (integer) sql Limit from.
 * Limitnumber (integer) maximum number of returned users.
 * sortby (string) sort by id, firstname or lastname. For ordering like the site does, use siteorder.
 * sortdirection (string) ASC or DESC

list of (

object {

name string //option name

value string //option value

}

)

XML-RPC (PHP structure)

```
[options] =>
    Array
    (
        [0] =>
            Array
            (
                [name] => string
                [value] => string
            )
    )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response

```

General structure

list of (
object {
id int //ID of the user
username string Optional //Username policy is defined in Moodle security config
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //An email address - allow email as root@localhost
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //An arbitrary ID code number perhaps from the institution
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
lastcourseaccess int Optional //last access to the course (0 if never)
description string Optional //User profile description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string Optional //User image profile URL - small version
profileimageurl string Optional //User image profile URL - big version
customfields Optional //User custom fields (also known as user profil fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
}
)groups Optional //user groups
list of (
object {
id int //group id
name string //group name
description string //group description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)roles Optional //user roles
list of (
object {
roleid int //role id
name string //role name
shortname string //role shortname
sortorder int //role sortorder
}
)preferences Optional //User preferences
list of (
object {
name string //The name of the preferences
value string //The value of the custom field
}
)enrolleddcourses Optional //Courses where the user is enrolled - Limited by which courses the user is able
list of (
object {
id int //Id of the course
fullname string //Fullname of the course
shortname string //Shortname of the course
}
)
)
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [lastcourseaccess] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => string
                                        [value] => string
                                        [name] => string
                                        [shortname] => string
                                    )
                        )
                )
            [groups] =>
                Array
                    (
                        [0] =>
                            Array
                                (
                                    [id] => int
                                    [name] => string
                                    [description] => string
                                    [descriptionformat] => int
                                )
                    )
            [roles] =>
                Array
                    (
                        [0] =>
                            Array
                                (
                                    [roleid] => int
                                    [name] => string
                                    [shortname] => string
                                    [sortorder] => int
                                )
                    )
            [preferences] =>
                Array
                    (
                        [0] =>
                            Array
                                (
                                    [name] => string
                                    [value] => string
                                )
                    )
            [enrolledcourses] =>
                Array
                    (
                        [0] =>
                            Array
                                (
                                    [id] => int
                                )
                    )
        )
)

```



```
[fullname] => string  
[shortname] => string  
)  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastcourseaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="country">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="customfields">
        <MULTIPLE>
          <SINGLE>
            <KEY name="type">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```



```
</KEY>
<KEY name="shortname">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="groups">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="roles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="roleid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="sortorder">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="enrolledcourses">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_enrol_get_enrolled_users_with_capability

For each course and capability specified, return a list of the users that are enrolled in the course and have that capability

Arguments**coursecapabilities** (Required)

course id and associated capability name

General structure

```
//course id and associated capability name
list of (
object {
courseid int //Course ID number in the Moodle course table
capabilities list of (
string //Capability name, such as mod/forum:viewdiscussion
)})
```

XML-RPC (PHP structure)

```
[coursecapabilities] =>
    Array
        (
            [0] =>
                Array
                    (
                        [courseid] => int
                        [capabilities] =>
                            Array
                                (
                                    [0] => string
                                )
                    )
            )
        )
```

**REST (POST parameters)**

```
coursecapabilities[0][courseid]= int
coursecapabilities[0][capabilities][0]= string
```

options (Default to "Array ()")

Option names: * groupid (integer) return only users in this group id. Requires 'moodle/site:accessallgroups'. * onlyactive (integer) only users with active enrolments. Requires 'moodle/course:enrolreview'. * userfields ('string, string, ...') return only the values of these user fields. * limitfrom (integer) sql limit from. * limitnumber (integer) max number of users per course and capability.

```
General structure

Default to "Array
(
)
" //Option names:
    * groupid (integer) return only users in this group id. Requires 'moodle/site:accessallgroups' .
    * onlyactive (integer) only users with active enrolments. Requires 'moodle/course:enrolreview' .
    * userfields ('string, string, ...') return only the values of these user fields.
    * limitfrom (integer) sql limit from.
    * limitnumber (integer) max number of users per course and capability.

list of (
object {
name string //option name
value string //option value
}
)
```

XML-RPC (PHP structure)

```
[options] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response

```
General structure

list of (
object {
courseid int //Course ID number in the Moodle course table
capability string //Capability name
users //List of users that are enrolled in the course and have the specified capability
list of (
object {
id int //ID of the user
username string Optional //Username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //Email address
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
lastcourseaccess int Optional //last access to the course (0 if never)
description string Optional //User profile description
descriptionformat int Optional //User profile description format
city string Optional //Home city of the user
country string Optional //Country code of the user, such as AU or CZ
profileimageurlsmall string Optional //User image profile URL - small
profileimageurl string Optional //User image profile URL - big
customfields Optional //User custom fields (also known as user profil fields)
list of (
object {
type string //The type of the custom field
value string //The value of the custom field
name string //The name of the custom field
shortname string //The shortname of the custom field
}
})groups Optional //user groups
list of (
object {
id int //group id
name string //group name
description string //group description
}
})roles Optional //user roles
list of (
object {
roleid int //role id
name string //role name
shortname string //role shortname
sortorder int //role sortorder
}
})preferences Optional //User preferences
list of (
object {
name string //The name of the preferences
value string //The value of the custom field
}
})enrolledcourses Optional //Courses where the user is enrolled - Limited by which courses the user is able to see
list of (
object {
id int //Id of the course
fullname string //Fullname of the course
shortname string //Shortname of the course
}
))
))
))
```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [courseid] => int
                [capability] => string
                [users] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [id] => int
                                        [username] => string
                                        [firstname] => string
                                        [lastname] => string
                                        [fullname] => string
                                        [email] => string
                                        [address] => string
                                        [phone1] => string
                                        [phone2] => string
                                        [department] => string
                                        [institution] => string
                                        [interests] => string
                                        [firstaccess] => int
                                        [lastaccess] => int
                                        [lastcourseaccess] => int
                                        [description] => string
                                        [descriptionformat] => int
                                        [city] => string
                                        [country] => string
                                        [profileimageurlsmall] => string
                                        [profileimageurl] => string
                                        [customfields] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [type] => string
                                                                [value] => string
                                                                [name] => string
                                                                [shortname] => string
                                                            )
                                                )
                                            )
                                        [groups] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [id] => int
                                                                [name] => string
                                                                [description] => string
                                                            )
                                                )
                                            )
                                        [roles] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [roleid] => int
                                                                [name] => string
                                                                [shortname] => string
                                                                [sortorder] => int
                                                            )
                                                )
                                            )
                                        [preferences] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [name] => string
                                                                [value] => string
                                                            )
                                                )
                                            )
                                    )
                                )
                            )
                        )
                    )
                [enrolledcourses] =>
            )
)

```



```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                )
            )
    ]
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="courseid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="capability">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="users">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="username">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="firstname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="lastname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="fullname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="email">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="address">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="phone1">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="phone2">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="department">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="institution">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="interests">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="firstaccess">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="lastaccess">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="lastcourseaccess">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="description">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="city">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="country">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimageurlsmall">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimageurl">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="customfields">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="type">
                    <VALUE>string</VALUE>

```



```
</KEY>
<KEY name="value">
    <VALUE>string</VALUE>
</KEY>
<KEY name="name">
    <VALUE>string</VALUE>
</KEY>
<KEY name="shortname">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="groups">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="roles">
        <MULTIPLE>
            <SINGLE>
                <KEY name="roleid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="shortname">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="sortorder">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="preferences">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
                </MULTIPLE>
            </KEY>
            <KEY name="enrolledcourses">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="id">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="fullname">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="shortname">
                            <VALUE>string</VALUE>
                        </KEY>
                    </SINGLE>
                    </MULTIPLE>
                </KEY>
                <KEY name=">
                    <SINGLE>
                    </MULTIPLE>
                </KEY>
            </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_enrol_get_potential_users

Get the list of potential users to enrol

Arguments

courseid (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

enrolid (Required)

enrolment id



General structure

```
int //enrolment id
```

XML-RPC (PHP structure)

```
[enrolid] => int
```

REST (POST parameters)

```
enrolid= int
```

search (Required)

query

General structure

```
string //query
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

searchanywhere (Required)

find a match anywhere, or only at the beginning

General structure

```
int //find a match anywhere, or only at the beginning
```

XML-RPC (PHP structure)

```
[searchanywhere] => int
```

REST (POST parameters)

```
searchanywhere= int
```

page (Required)

Page number

General structure

```
int //Page number
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

**perpage** (Required)

Number per page

General structure

```
int //Number per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

Response

```
General structure

list of (
object {
id int    //ID of the user
username string Optional //The username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //An email address - allow email as root@localhost
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //An arbitrary ID code number perhaps from the institution
interests string Optional //User interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
auth string Optional //Auth plugins include manual, Ldap, etc
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
lang string Optional //Language code such as "en", must exist on server
calendartype string Optional //Calendar type such as "gregorian", must exist on server
theme string Optional //Theme name such as "standard", must exist on server
timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
description string Optional //User profile description
descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string //User image profile URL - small version
profileimageurl string //User image profile URL - big version
customfields Optional //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field (as stored in the database)
displayvalue string Optional //The value of the custom field for display
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
})
preferences Optional //Users preferences
list of (
object {
name string //The name of the preferences
value string //The value of the preference
})
})
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [suspended] => int
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => string
                                        [value] => string
                                        [displayvalue] => string
                                        [name] => string
                                        [shortname] => string
                                    )
                        )
                [preferences] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [value] => string
                                    )
                        )
                )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="auth">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="suspended">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="confirmed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="calendartype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="theme">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timezone">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="mailformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
```



```

</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_enrol_get_users_courses

Get the list of courses where a user is enrolled in

Arguments

userid (Required)

user id

General structure

```
int //user id
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

returnusercount (Default to "1")

Include count of enrolled users for each course? This can add several seconds to the response time if a user is on several large courses, so set this to false if the value will not be used to improve performance.

General structure

```
int Default to "1" /Include count of enrolled users for each course? This can add several seconds to the response time if a user is on several large courses, so set this to false if the value will not be used to improve performance.
```

XML-RPC (PHP structure)

```
[returnusercount] => int
```

REST (POST parameters)

```
returnusercount= int
```

Response



General structure

```

list of (
object {
id int //id of course
shortname string //short name of course
fullname string //long name of course
displayname string Optional //course display name for lists.
enrolledusercount int Optional //Number of enrolled users in this course
idnumber string //id number of course
visible int //1 means visible, 0 means not yet visible course
summary string Optional //summary
summaryformat int Optional //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
format string Optional //course format: weeks, topics, social, site
courseimage string Optional //The course image URL
showgrades int Optional //true if grades are shown, otherwise false
lang string Optional //forced course language
enablecompletion int Optional //true if completion is enabled, otherwise false
completionhascriteria int Optional //If completion criteria is set.
completionusertracked int Optional //If the user is completion tracked.
category int Optional //course category id
progress double Optional //Progress percentage
completed int Optional //Whether the course is completed.
startdate int Optional //Timestamp when the course start
enddate int Optional //Timestamp when the course end
marker int Optional //Course section marker.
lastaccess int Optional //Last access to the course (timestamp).
isfavourite int Optional //If the user marked this course a favourite.
hidden int Optional //If the user hide the course from the dashboard.
overviewfiles Optional //Overview files attached to this course.
list of (
  //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)showactivitydates int //Whether the activity dates are shown or not
showcompletionconditions int //Whether the activity completion conditions are shown or not
timemodified int Optional //Last time course settings were updated (timestamp).
}
)

```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [shortname] => string
                [fullname] => string
                [displayname] => string
                [enrolledusercount] => int
                [idnumber] => string
                [visible] => int
                [summary] => string
                [summaryformat] => int
                [format] => string
                [courseimage] => string
                [showgrades] => int
                [lang] => string
                [enablecompletion] => int
                [completionhascriteria] => int
                [completionusertracked] => int
                [category] => int
                [progress] => double
                [completed] => int
                [startdate] => int
                [enddate] => int
                [marker] => int
                [lastaccess] => int
                [isfavourite] => int
                [hidden] => int
                [overviewfiles] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [filename] => string
                                        [filepath] => string
                                        [filesize] => int
                                        [fileurl] => string
                                        [timemodified] => int
                                        [mimetype] => string
                                        [isexternalfile] => int
                                        [repositorytype] => string
                                    )
                        )
                [showactivitydates] => int
                [showcompletionconditions] => int
                [timemodified] => int
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="shortname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="displayname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="enrolledusercount">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="visible">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="summary">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="summaryformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="format">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="courseimage">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="showgrades">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="enablecompletion">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionhascriteria">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="completionusertracked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="category">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="progress">
        <VALUE>double</VALUE>
      </KEY>
      <KEY name="completed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="startdate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enddate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="marker">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isfavourite">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="hidden">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="overviewfiles">
        <MULTIPLE>
```



```

<SINGLE>
    <KEY name="filename">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filepath">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filesize">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="fileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="timemodified">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="mimetype">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="isexternalfile">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="repositorytype">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="showactivitydates">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_enrol_search_users

Search within the list of course participants

Arguments

courseid (Required)

course id

General structure

```
int //course id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

search (Required)

query

General structure

```
string //query
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

searchanywhere (Required)

find a match anywhere, or only at the beginning

General structure

```
int //find a match anywhere, or only at the beginning
```

XML-RPC (PHP structure)

```
[searchanywhere] => int
```

REST (POST parameters)

```
searchanywhere= int
```

**page** (Required)

Page number

General structure

```
int //Page number
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Required)

Number per page

```
General structure

int //Number per page
```

```
XML-RPC (PHP structure)

[perpage] => int
```

```
REST (POST parameters)

perpage= int
```

Response

```
General structure

list of (
object {
id int //ID of the user
username string Optional //The username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //An email address - allow email as root@localhost
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //An arbitrary ID code number perhaps from the institution
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //Last access to the site (0 if never)
auth string Optional //Auth plugins include manual, ldap, etc
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
lang string Optional //Language code such as "en", must exist on server
calendartype string Optional //Calendar type such as "gregorian", must exist on server
theme string Optional //Theme name such as "standard", must exist on server
timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
description string Optional //User profile description
descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string //User image profile URL - small version
profileimageurl string //User image profile URL - big version
customfields Optional //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field (as stored in the database)
displayvalue string Optional //The value of the custom field for display
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
})
preferences Optional //Users preferences
list of (
object {
name string //The name of the preferences
value string //The value of the preference
})
)
)
```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [suspended] => int
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => string
                                        [value] => string
                                        [displayvalue] => string
                                        [name] => string
                                        [shortname] => string
                                    )
                        )
                [preferences] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [value] => string
                                    )
                        )
                )
            )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="auth">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="suspended">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="confirmed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="calendartype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="theme">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timezone">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="mailformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
```



```

</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

[core_enrol_submit_user_enrolment_form](#)

Submit form data for enrolment form

Arguments

formdata (Required)

The data from the event form

General structure

```
string //The data from the event form
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response

General structure

```
object {
    result int //True if the user's enrolment was successfully updated
    validationerror int Default to "" //Indicates invalid form data
}
```

XML-RPC (PHP structure)

```
Array
(
    [result] => int
    [validationerror] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="validationerror">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_enrol_unenrol_user_enrolment

External function that unenrolls a given user enrolment

Arguments**ueid** (Required)

User enrolment ID

General structure

```
int //User enrolment ID
```

XML-RPC (PHP structure)

```
[ueid] => int
```

REST (POST parameters)

```
ueid= int
```

Response**General structure**

```
object {
  result int //True if the user's enrolment was successfully updated
  errors //List of validation errors
  list of (
    object {
      key string //The data that failed the validation
      message string //The error message
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
  [result] => int
  [errors] =>
    Array
    (
      [0] =>
        Array
        (
          [key] => string
          [message] => string
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="errors">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="key">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_fetch_notifications

Return a list of notifications for the current session

**Arguments****contextid** (Required)

Context ID

General structure

int //Context ID

XML-RPC (PHP structure)

[contextid] => int

REST (POST parameters)

contextid= int

Response**General structure**

```
list of (
object {
template string //Name of the template
variables object {
message string //HTML content of the Notification
extraclasses string //Extra classes to provide to the template
announce string //Whether to announce
closebutton string //Whether to close
}
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
    (
        [
            [template] => string
            [variables] =>
                Array
                (
                    [
                        [message] => string
                        [extraclasses] => string
                        [announce] => string
                        [closebutton] => string
                    )
                )
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="template">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="variables">
                <SINGLE>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="extraclasses">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="announce">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="closebutton">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_files_delete_draft_files

Delete the indicated files (or directories) from a user draft file area.

Arguments

draftitemid (Required)

Item id of the draft file area

General structure

```
int //Item id of the draft file area
```

XML-RPC (PHP structure)

```
[draftitemid] => int
```

REST (POST parameters)

```
draftitemid= int
```

files (Required)

Files or directories to be deleted.

General structure

```
//Files or directories to be deleted.
list of (
object {
filepath string //Path to the file or directory to delete.
filename string //Name of the file to delete.
}
)
```

XML-RPC (PHP structure)

```
[files] =>
  Array
  (
    [0] =>
      Array
      (
        [filepath] => string
        [filename] => string
      )
  )
```

REST (POST parameters)

```
files[0][filepath]= string
files[0][filename]= string
```

Response

General structure

```

object {
parentpaths list of (
string //Path to parent directory of the deleted files.
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)

```

XML-RPC (PHP structure)

```

Array
(
    [parentpaths] =>
        Array
            (
                [0] => string
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="parentpaths">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_files_get_files

browse moodle files

Arguments**contextid** (Required)

context id Set to -1 to use contextlevel and instanceid.

General structure

```
int //context id Set to -1 to use contextlevel and instanceid.
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

component (Required)

component

General structure

```
string //component
```

**XML-RPC (PHP structure)**

```
[component] => string
```

REST (POST parameters)

```
component= string
```

filearea (Required)

file area

General structure

```
string //file area
```

XML-RPC (PHP structure)

```
[filearea] => string
```

REST (POST parameters)

```
filearea= string
```

itemid (Required)

associated id

General structure

```
int //associated id
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

filepath (Required)

file path

General structure

```
string //file path
```

XML-RPC (PHP structure)

```
[filepath] => string
```

REST (POST parameters)

```
filepath= string
```

filename (Required)

file name

General structure

```
string //file name
```

XML-RPC (PHP structure)

```
[filename] => string
```

REST (POST parameters)

```
filename= string
```

modified (Default to "null")

timestamp to return files changed after this time.

General structure

```
int Default to "null" //timestamp to return files changed after this time.
```

XML-RPC (PHP structure)

```
[modified] => int
```

REST (POST parameters)

```
modified= int
```

contextlevel (Default to "null")

The context level for the file location.

General structure

```
string Default to "null" //The context level for the file location.
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Default to "null")

The instance id for where the file is located.

General structure

```
int Default to "null" //The instance id for where the file is located.
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

Response**General structure**

```
object {
  parents list of (
    object {
      contextid int component string filearea string itemid int filepath string filename string }
    )files list of (
      object {
        contextid int component string filearea string itemid int filepath string filename string isdir int url string timemodified int timecreated int Optional
        //Time created
        filesize int Optional //File size
        author string Optional //File owner
        license string Optional //File license
      }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [parents] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [contextid] => int
                            [component] => string
                            [filearea] => string
                            [itemid] => int
                            [filepath] => string
                            [filename] => string
                        )
                )
            )
    [files] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [contextid] => int
                            [component] => string
                            [filearea] => string
                            [itemid] => int
                            [filepath] => string
                            [filename] => string
                            [isdir] => int
                            [url] => string
                            [timemodified] => int
                            [timecreated] => int
                            [filesize] => int
                            [author] => string
                            [license] => string
                        )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="parents">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="contextid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filearea">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="filepath">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filename">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="files">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="contextid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filearea">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="filepath">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isdir">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="url">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="filesize">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="author">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="license">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_files_get_unused_draft_itemid

Generate a new draft itemid for the current user.

Arguments**Response****General structure**

```
object {
    component string //File area component.
    contextid int //File area context.
    userid int //File area user id.
    filearea string //File area name.
    itemid int //File are item id.
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [component] => string
    [contextid] => int
    [userid] => int
    [filearea] => string
    [itemid] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="filearea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_files_upload

upload a file to moodle

Arguments

contextid (Default to "null")

context id

General structure

```
int Default to "null" //context id
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

filearea (Required)

file area

General structure

```
string //file area
```

XML-RPC (PHP structure)

```
[filearea] => string
```

REST (POST parameters)

```
filearea= string
```



itemid (Required)

associated id

General structure

```
int //associated id
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

filepath (Required)

file path

General structure

```
string //file path
```

XML-RPC (PHP structure)

```
[filepath] => string
```

REST (POST parameters)

```
filepath= string
```

filename (Required)

file name

General structure

```
string //file name
```

XML-RPC (PHP structure)

```
[filename] => string
```

REST (POST parameters)

```
filename= string
```

filecontent (Required)

file content

General structure

```
string //file content
```

XML-RPC (PHP structure)

```
[filecontent] => string
```

REST (POST parameters)

```
filecontent= string
```



contextlevel (Default to "null")

The context level to put the file in, (block, course, coursecat, system, user, module)

General structure

```
string Default to "null" //The context level to put the file in,
(block, course, coursecat, system, user, module)
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Default to "null")

The Instance id of item associated with the context level

General structure

```
int Default to "null" //The Instance id of item associated
with the context Level
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

Response**General structure**

```
object {
    contextid int component string filearea string itemid int filepath string filename string url string }
```

XML-RPC (PHP structure)

```
Array
(
    [contextid] => int
    [component] => string
    [filearea] => string
    [itemid] => int
    [filepath] => string
    [filename] => string
    [url] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="filearea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="filepath">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="filename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="url">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_filters_get_available_in_context

Returns the filters available in the given contexts.

Arguments**contexts** (Required)

The list of contexts to check.

General structure

```
//The List of contexts to check.
list of (
object {
contextlevel string //The context Level where the filters are:
                    (coursecat, course, module)
instanceid int //The instance id of item associated with the context.
})
```

XML-RPC (PHP structure)

```
[contexts] =>
  Array
  (
  [0] =>
    Array
    (
      [contextlevel] => string
      [instanceid] => int
    )
  )
  ^
```

REST (POST parameters)

```
contexts[0][contextlevel]= string
contexts[0][instanceid]= int
```

Response

```
General structure

object {
filters //Available filters
list of (
object {
contextlevel string //The context level where the filters are:
(cOURSECAT, course, module).
instanceid int //The instance id of item associated with the context.
contextid int //The context id.
filter string //Filter plugin name.
localstate int //Filter state: 1 for on, -1 for off, 0 if inherit.
inheritedstate int //1 or 0 to use when Localstate is set to inherit.
}
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```

XML-RPC (PHP structure)

```
Array
(
[filters] =>
Array
(
[0] =>
Array
(
[contextlevel] => string
[instanceid] => int
[contextid] => int
[filter] => string
[localstate] => int
[inheritedstate] => int
)
)
)
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="filters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="contextlevel">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="instanceid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="contextid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="filter">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="localstate">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="inheritedstate">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_form_dynamic_form

Process submission of a dynamic (modal) form

Arguments

form (Required)

Form class

General structure

```
string //Form class
```

XML-RPC (PHP structure)

```
[form] => string
```

REST (POST parameters)

```
form= string
```

formdata (Required)

url-encoded form data

General structure

```
string //url-encoded form data
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response

General structure

```
object {
  submitted int //If form was submitted and validated
  data string Optional //JSON-encoded return data from form processing method
  html string Optional //HTML fragment of the form
  javascript string Optional //JavaScript fragment of the form
}
```



XML-RPC (PHP structure)

```
Array
(
  [submitted] => int
  [data] => string
  [html] => string
  [javascript] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="submitted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="data">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="html">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="javascript">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_form_get_filetypes_browser_data

Provides data for the filetypes element browser.

Arguments**onlytypes** (Default to "")

Limit the browser to the given groups and extensions

General structure

```
string Default to "" //Limit the browser to the given groups and extensions
```

XML-RPC (PHP structure)

```
[onlytypes] => string
```

REST (POST parameters)

```
onlytypes= string
```

allowall (Default to "1")

Allows to select All file types, does not apply with onlytypes are set.

General structure

```
int Default to "1" //Allows to select ALL file types, does not apply with onlytypes are set.
```

XML-RPC (PHP structure)

```
[allowall] => int
```

REST (POST parameters)

```
allowall= int
```

current(Default to "")

Current types that should be selected.

General structure

```
string Default to "" //Current types that should be selected.
```

XML-RPC (PHP structure)

```
[current] => string
```

REST (POST parameters)

```
current= string
```

Response**General structure**

```
object {
groups //List of file type groups
list of (
object {
key string //The file type group identifier
name string //The file type group name
selectable int //Can it be marked as selected
selected int //Should it be marked as selected
ext string //The list of file extensions associated with the group
expanded int //Should the group start as expanded or collapsed
types //List of file types in the group
list of (
object {
key string //The file type identifier
name string //The file type name
selected int //Should it be marked as selected
ext string //The file extension associated with the file type
}
)
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [groups] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [key] => string
                            [name] => string
                            [selectable] => int
                            [selected] => int
                            [ext] => string
                            [expanded] => int
                            [types] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [key] => string
                                                    [name] => string
                                                    [selected] => int
                                                    [ext] => string
                                                )
                                    )
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groups">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="key">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="selectable">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="selected">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="ext">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="expanded">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="types">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="key">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="selected">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="ext">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_get_component_strings

Return all raw strings (with {\$a->xxx}), for a specific component - similar to core_get_component_strings(), call

Arguments

component (Required)
component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

lang (Default to "null")

```
lang
```

General structure

```
string Default to "null" //Lang
```

XML-RPC (PHP structure)

```
[lang] => string
```

REST (POST parameters)

```
lang= string
```

Response

General structure

```
list of (
object {
stringid string //string id
string string //translated string
}
)
```



XML-RPC (PHP structure)

```
Array
(
[0] =>
    Array
    (
        [stringid] => string
        [string] => string
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="stringid">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="string">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_get_fragment

Return a fragment for inclusion, such as a JavaScript page.

Arguments**component** (Required)

Component for the callback e.g. mod_assign

**General structure**

```
string //Component for the callback e.g. mod_assign
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

callback (Required)

Name of the callback to execute

General structure

```
string //Name of the callback to execute
```

XML-RPC (PHP structure)

```
[callback] => string
```

REST (POST parameters)

```
callback= string
```

contextid (Required)

Context ID that the fragment is from

General structure

```
int //Context ID that the fragment is from
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

args (Optional)

args for the callback are optional

General structure

```
Optional //args for the callback are optional
list of (
object {
name string //param name
value string //param value
}
)
```

XML-RPC (PHP structure)

```
[args] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

**REST (POST parameters)**

```
args[0][name]= string
args[0][value]= string
```

Response

General structure

```
object {
    html string //HTML fragment.
    javascript string //JavaScript fragment
}
```

XML-RPC (PHP structure)

```
Array
(
    [html] => string
    [javascript] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="html">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="javascript">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

Yes

core_get_string

Return a translated string - similar to core get_string(), call

Arguments**stringid** (Required)

string identifier

General structure

```
string //string identifier
```

XML-RPC (PHP structure)

```
[stringid] => string
```

REST (POST parameters)

```
stringid= string
```

component (Default to "moodle")

component

General structure

```
string Default to "moodle" //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

lang (Default to "null")

lang

General structure

```
string Default to "null" //Lang
```

XML-RPC (PHP structure)

```
[lang] => string
```

REST (POST parameters)

```
lang= string
```

stringparams (Default to "Array ()")

the definition of a string param (i.e. {\$a->name})

General structure

```
Default to "Array
(
)
" //the definition of a string param (i.e. {$a->name})
list of (
object {
name string Optional //param name
          - if the string expect only one $a parameter then don't send this field, just send the value.
value string //param value
}
)
```

XML-RPC (PHP structure)

```
[stringparams] =>
    Array
        (
            [
                [0] =>
                    Array
                        (
                            [name] => string
                            [value] => string
                        )
            )
        )
    )
```

REST (POST parameters)

```
stringparams[0][name]= string
stringparams[0][value]= string
```

Response

translated string

General structure

```
string //translated string
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

Callable from AJAX

Yes

core_get_strings

Return some translated strings - like several core get_string() calls

Arguments

strings (Required)

General structure

```
list of (
object {
stringid string //string identifier
component string Default to "moodle" //component
lang string Default to "null" //lang
stringparams Default to "Array"
(
)
" //the definition of a string param (i.e. {$a->name})
list of (
object {
name string Optional //param name
value string //param value
}
)}
)
```

XML-RPC (PHP structure)

```
[strings] =>
  Array
    (
      [0] =>
        Array
          (
            [stringid] => string
            [component] => string
            [lang] => string
            [stringparams] =>
              Array
                (
                  [0] =>
                    Array
                      (
                        [name] => string
                        [value] => string
                      )
                )
            )
        )
    )
)
```

REST (POST parameters)

```
strings[0][stringid]= string
strings[0][component]= string
strings[0][lang]= string
strings[0][stringparams][0][name]= string
strings[0][stringparams][0][value]= string
```



Response

General structure

```
list of (
object {
stringid string //string id
component string //string component
lang string //lang
string string //translated string
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [stringid] => string
                    [component] => string
                    [lang] => string
                    [string] => string
                ]
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="stringid">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="lang">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="string">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

Callable from AJAX

Yes

core_get_user_dates

Return formatted timestamps

Arguments**contextid** (Default to "0")

Context ID. Either use this value, or level and instanceid.

General structure

```
int Default to "0" //Context ID. Either use this value, or Level and instanceid.
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

contextlevel (Default to "")

Context level. To be used with instanceid.

General structure

```
string Default to "" //Context Level. To be used with instanceid.
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Default to "0")

Context instance ID. To be used with level

General structure

```
int Default to "0" //Context instance ID. To be used with level
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```



timestamps (Required)

General structure

```
list of (
object {
timestamp int //unix timestamp
format string //format string
type string Default to "null" //The calendar type
fixday int Default to "1" //Remove Leading zero for day
fixhour int Default to "1" //Remove Leading zero for hour
})
```

XML-RPC (PHP structure)

```
[timestamps] =>
    Array
        (
            [0] =>
                Array
                    (
                        [timestamp] => int
                        [format] => string
                        [type] => string
                        [fixday] => int
                        [fixhour] => int
                    )
        )
)
```

REST (POST parameters)

```
timestamps[0][timestamp]= int
timestamps[0][format]= string
timestamps[0][type]= string
timestamps[0][fixday]= int
timestamps[0][fixhour]= int
```

Response**General structure**

```
object {
    dates list of (
        string //formatted dates strings
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [dates] =>
        Array
            (
                [0] => string
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="dates">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_create_gradecategories

Create grade categories inside a course gradebook.

Arguments

courseid (Required)

id of course

General structure

```
int //id of course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

categories (Required)

Categories to create

General structure

```
//Categories to create
list of (
    //Category to create
    object {
        fullname string //fullname of category
        options Default to "Array"
        (
        )
        " //optional category data
        object {
            aggregation int Optional //aggregation method
            aggregateonlygraded int Optional //exclude empty grades
            aggregateoutcomes int Optional //aggregate outcomes
            droplow int Optional //drop Low grades
            itemname string Optional //the category total name
            iteminfo string Optional //the category iteminfo
            idnumber string Optional //the category idnumber
            gradetype int Optional //the grade type
            grademax int Optional //the grade max
            grademin int Optional //the grade min
            gradepass int Optional //the grade to pass
            display int Optional //the display type
            decimals int Optional //the decimal count
            hiddenuntil int Optional //grades hidden until
            locktime int Optional //Lock grades after
            weightoverride int Optional //weight adjusted
            aggregationcoef2 string Optional //weight coefficient
            parentcategoryid int Optional //The parent category id
            parentcategoryidnumber string Optional //the parent category idnumber
        }
    }
)
```

XML-RPC (PHP structure)

```
[categories] =>
  Array
    (
      [0] =>
        Array
          (
            [fullname] => string
            [options] =>
              Array
                (
                  [aggregation] => int
                  [aggregateonlygraded] => int
                  [aggregateoutcomes] => int
                  [droplow] => int
                  [itemname] => string
                  [iteminfo] => string
                  [idnumber] => string
                  [gradetype] => int
                  [grademax] => int
                  [grademin] => int
                  [gradepass] => int
                  [display] => int
                  [decimals] => int
                  [hiddenuntil] => int
                  [locktime] => int
                  [weightoverride] => int
                  [aggregationcoef2] => string
                  [parentcategoryId] => int
                  [parentcategoryIdnumber] => string
                )
          )
    )
)
```

REST (POST parameters)

```
categories[0][fullname]= string
categories[0][options][aggregation]= int
categories[0][options][aggregateonlygraded]= int
categories[0][options][aggregateoutcomes]= int
categories[0][options][droplow]= int
categories[0][options][itemname]= string
categories[0][options][iteminfo]= string
categories[0][options][idnumber]= string
categories[0][options][gradetype]= int
categories[0][options][grademax]= int
categories[0][options][grademin]= int
categories[0][options][gradepass]= int
categories[0][options][display]= int
categories[0][options][decimals]= int
categories[0][options][hiddenuntil]= int
categories[0][options][locktime]= int
categories[0][options][weightoverride]= int
categories[0][options][aggregationcoef2]= string
categories[0][options][parentcategoryid]= int
categories[0][options][parentcategoryidnumber]= string
```



Response

General structure

```
object {
categoryids list of (
int //created category ID
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)})
```

XML-RPC (PHP structure)

```
Array
(
    [categoryids] =>
        Array
        (
            [
                [0] => int
            )
        )
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="categoryids">
            <MULTIPLE>
                <VALUE>int</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_grades_create_gradecategory (Deprecated)

** DEPRECATED ** Please do not call this function any more. Use core_grades_create_gradecategories. Create a grade category inside a course gradebook.

Arguments

courseid (Required)

id of course

General structure

```
int //id of course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

fullname (Required)

fullname of category

General structure

```
string //fullname of category
```

XML-RPC (PHP structure)

```
[fullname] => string
```

REST (POST parameters)

```
fullname= string
```

options (Default to "Array ()")

optional category data



General structure

```

Default to "Array
(
)
" //optional category data
object {
aggregation int  Optional //aggregation method
aggregateonlygraded int  Optional //exclude empty grades
aggregateoutcomes int  Optional //aggregate outcomes
droplow int  Optional //drop low grades
itemname string  Optional //the category total name
iteminfo string  Optional //the category iteminfo
idnumber string  Optional //the category idnumber
gradetype int  Optional //the grade type
grademax int  Optional //the grade max
grademin int  Optional //the grade min
gradepass int  Optional //the grade to pass
display int  Optional //the display type
decimals int  Optional //the decimal count
hiddenuntil int  Optional //grades hidden until
locktime int  Optional //lock grades after
weightoverride int  Optional //weight adjusted
aggregationcoef2 string  Optional //weight coefficient
parentcategoryid int  Optional //The parent category id
parentcategoryidnumber string  Optional //the parent category idnumber
}

```

XML-RPC (PHP structure)

```
[options] =>
Array
(
[aggregation] => int
[aggregateonlygraded] => int
[aggregateoutcomes] => int
[droplow] => int
[itemname] => string
[iteminfo] => string
[idnumber] => string
[gradetype] => int
[grademax] => int
[grademin] => int
[gradepass] => int
[display] => int
[decimals] => int
[hiddenuntil] => int
[locktime] => int
[weightoverride] => int
[aggregationcoef2] => string
[parentcategoryid] => int
[parentcategoryidnumber] => string
)
```

**REST (POST parameters)**

```
options[aggregation]= int
options[aggregateonlygraded]= int
options[aggregateoutcomes]= int
options[droplow]= int
options[itemname]= string
options[iteminfo]= string
options[idnumber]= string
options[gradetype]= int
options[grademax]= int
options[grademin]= int
options[gradepass]= int
options[display]= int
options[decimals]= int
options[hiddenuntil]= int
options[locktime]= int
options[weightoverride]= int
options[aggregationcoef2]= string
options[parentcategoryid]= int
options[parentcategoryidnumber]= string
```

Response

General structure

```
object {
    categoryid int //The ID of the created category
    warnings Optional //list of warnings
    list of (
        /warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [categoryid] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="categoryid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_grades_get_enrolled_users_for_search_widget ([Deprecated](#))

**** DEPRECATED **** Please do not call this function any more. Use core_grades_get_enrolled_users_for_selector instead. Returns the enrolled users within and map some fields to the returned array of user objects.

Arguments**courseid** (Required)

Course Id

General structure

```
int //Course Id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

actionbaseurl (Required)

The base URL for the user option

General structure

```
string //The base URL for the user option
```

XML-RPC (PHP structure)

```
[actionbaseurl] => string
```

REST (POST parameters)

```
actionbaseurl= string
```

groupid (Default to "0")

Group Id

General structure

```
int Default to "0" //Group Id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response

General structure

```
object {
    users list of (
        object {
            id int //ID of the user
            profileimage string Optional //The Location of the users larger image
            url string Optional //The Link to the user report
            fullname string Optional //The full name of the user
            email string Optional //An email address - allow email as root@localhost
            active int //Are we currently on this item?
        }
    )warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [users] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [profileimage] => string
                        [url] => string
                        [fullname] => string
                        [email] => string
                        [active] => int
                    ]
                )
            )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    ]
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="profileimage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="url">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="email">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="active">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_get_enrolled_users_for_selector

Returns the enrolled users within and map some fields to the returned array of user objects.

Arguments

courseid (Required)

Course Id

General structure

```
int //Course Id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

groupid (Default to "0")

GroupId

General structure

```
int Default to "0" //Group Id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response

General structure

```
object {
  users list of (
    object {
      id int //ID of the user
      profileimage string Optional //The location of the users larger image
      fullname string Optional //The full name of the user
      email string Optional //An email address - allow email as root@localhost
    }
    )warnings Optional //List of warnings
    list of (
      //warning
      object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
      }
    )
  )
}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [users] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [profileimage] => string
                                [fullname] => string
                                [email] => string
                            )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="profileimage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="email">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_get_feedback

Get the feedback data for a grade item

Arguments

courseid (Required)

Course ID

General structure

```
int //Course ID
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Required)

User ID

General structure

```
int //User ID
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

itemid (Required)

Grade Item ID

General structure

```
int //Grade Item ID
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

Response**General structure**

```
object {
    feedbacktext string //The full feedback text
    title string //Title of the grade item that the feedback is for
    fullname string //Students name
    picture string //Students picture
    additionalfield string //Additional field for the user (email or ID number, for example)
}
```

XML-RPC (PHP structure)

```
Array
(
    [feedbacktext] => string
    [title] => string
    [fullname] => string
    [picture] => string
    [additionalfield] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="feedbacktext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="title">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="picture">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="additionalfield">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Yes

Callable from AJAX

Yes

core_grades_get_gradeitems

Get the gradeitems for a course

Arguments

courseid (Required)

Course ID

General structure

```
int //Course ID
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
object {
    gradeItems list of (
        object {
            id string //An ID for the grade item
            itemname string //The full name of the grade item
            category string Optional //The grade category of the grade item
        }
        )warnings Optional //list of warnings
        list of (
            //warning
            object {
                item string Optional //item
                itemid int Optional //item id
                warningcode string //the warning code can be used by the client app to implement specific behaviour
                message string //untranslated english message to explain the warning
            }
        )
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [gradeItems] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => string
                    [itemname] => string
                    [category] => string
                )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="gradeItems">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="category">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    <SINGLE>
        <KEY>
    </RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_get_groups_for_search_widget

**** DEPRECATED **** Please do not call this function any more. Use core_grades_get_groups_for_selector instead. Get the group/(s) for a course

Arguments**courseid** (Required)

Course Id

General structure

int //Course Id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

Response**General structure**

```
object {
groups list of (
object {
id string //An ID for the group
name string //The full name of the group
})
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```

XML-RPC (PHP structure)

```
Array
(
    [groups] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => string
                    [name] => string
                )
            )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groups">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_get_groups_for_selector

Get the group/(s) for a course

Arguments**courseid** (Required)

Course Id

General structure

int //Course Id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

Response**General structure**

```
object {
groups list of (
object {
id string //An ID for the group
name string //The full name of the group
})
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```

XML-RPC (PHP structure)

```
Array
(
    [groups] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => string
                    [name] => string
                )
            )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groups">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_grader_gradingpanel_point_fetch

Fetch the data required to display the grader grading panel for simple grading, creating the grade item if required

Arguments**component** (Required)

The name of the component

General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

Response**General structure**

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        grade double //The numeric grade
        usergrade string //Current user grade
        maxgrade string //Max possible grade
        gradedby string //The assumed grader of this grading instance
        timecreated int //The time that the grade was created
        timemodified int //The time that the grade was last updated
    }
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```



XML-RPC (PHP structure)

```
Array
(
    [templatename] => string
    [hasgrade] => int
    [grade] =>
        Array
        (
            [grade] => double
            [usergrade] => string
            [maxgrade] => string
            [gradedby] => string
            [timecreated] => int
            [timemodified] => int
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="templatename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="hasgrade">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
            <SINGLE>
                <KEY name="grade">
                    <VALUE>double</VALUE>
                </KEY>
                <KEY name="usergrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="maxgrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="gradedby">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_grader_gradingpanel_point_store

Store the data required to display the grader grading panel for simple grading

Arguments**component** (Required)

The name of the component

General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

**XML-RPC (PHP structure)**

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

notifyuser (Default to "")

Wheteher to notify the user or not

General structure

```
int Default to "" //Wheteher to notify the user or not
```

XML-RPC (PHP structure)

```
[notifyuser] => int
```

REST (POST parameters)

```
notifyuser= int
```

formdata (Required)

The serialised form data representing the grade

General structure

```
string //The serialised form data representing the grade
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response**General structure**

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        grade double //The numeric grade
        usergrade string //Current user grade
        maxgrade string //Max possible grade
        gradedby string //The assumed grader of this grading instance
        timecreated int //The time that the grade was created
        timemodified int //The time that the grade was last updated
    }
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    })
}
```

XML-RPC (PHP structure)

```
Array
(
    [templatename] => string
    [hasgrade] => int
    [grade] =>
        Array
        (
            [grade] => double
            [usergrade] => string
            [maxgrade] => string
            [gradedby] => string
            [timecreated] => int
            [timemodified] => int
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="templatename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="hasgrade">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
            <SINGLE>
                <KEY name="grade">
                    <VALUE>double</VALUE>
                </KEY>
                <KEY name="usergrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="maxgrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="gradedby">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_grades_grader_gradingpanel_scale_fetch

Fetch the data required to display the grader grading panel for scale-based grading, creating the grade item if required

Arguments

component (Required)

The name of the component

General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```



XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

[gradeduserid] => int

REST (POST parameters)

gradeduserid= int

Response

```
General structure

object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        options //The description of the grade option
        list of (
            object {
                value double //The grade value
                title string //The description fo the option
                selected int //Whether this item is currently selected
            }
        )
        usergrade string //Current user grade
        maxgrade string //Max possible grade
        gradedby string //The assumed grader of this grading instance
        timecreated int //The time that the grade was created
        timemodified int //The time that the grade was last updated
    }
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
})}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [templatename] => string
        [hasgrade] => int
        [grade] =>
            Array
            (
                [
                    [options] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [value] => double
                                            [title] => string
                                            [selected] => int
                                        )
                                    )
                            )
                ]
            )
        [usergrade] => string
        [maxgrade] => string
        [gradedby] => string
        [timecreated] => int
        [timemodified] => int
    )
)
[warnings] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="templatename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="hasgrade">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
            <SINGLE>
                <KEY name="options">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="value">
                                <VALUE>double</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="selected">
                                <VALUE>int</VALUE>
                            </KEY>
                        </SINGLE>
                    </MULTIPLE>
                </KEY>
                <KEY name="usergrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="maxgrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="gradedby">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_grades_grader_gradingpanel_scale_store`

Store the data required to display the grader grading panel for scale-based grading

Arguments

`component` (Required)

The name of the component

General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

`contextid` (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

`itemname` (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

notifyuser (Default to "")

Wheteher to notify the user or not

General structure

```
int Default to "" //Wheteher to notify the user or not
```

XML-RPC (PHP structure)

```
[notifyuser] => int
```

REST (POST parameters)

```
notifyuser= int
```

formdata (Required)

The serialised form data representing the grade

General structure

```
string //The serialised form data representing the grade
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response

General structure

```

object {
  templatename string //The template to use when rendering this data
  hasgrade int //Does the user have a grade?
  grade object {
    options //The description of the grade option
    list of (
    object {
      value double //The grade value
      title string //The description fo the option
      selected int //Whether this item is currently selected
    }
    )usergrade string //Current user grade
    maxgrade string //Max possible grade
    gradedby string //The assumed grader of this grading instance
    timecreated int //The time that the grade was created
    timemodified int //The time that the grade was last updated
  }
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}
)

```

XML-RPC (PHP structure)

```

Array
(
  [templatename] => string
  [hasgrade] => int
  [grade] =>
    Array
    (
      [options] =>
        Array
        (
          [0] =>
            Array
            (
              [value] => double
              [title] => string
              [selected] => int
            )
        )
      [usergrade] => string
      [maxgrade] => string
      [gradedby] => string
      [timecreated] => int
      [timemodified] => int
    )
  [warnings] =>
    Array
    (
      [0] =>
        Array
        (
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
    )
)
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="templatename">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="hasgrade">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="grade">
            <SINGLE>
                <KEY name="options">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="value">
                                <VALUE>double</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="selected">
                                <VALUE>int</VALUE>
                            </KEY>
                        </SINGLE>
                    </MULTIPLE>
                </KEY>
                <KEY name="usergrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="maxgrade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="gradedby">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_grades_update_grades`

Update a grade item and associated student grades.

Arguments

`source` (Required)

The source of the grade update

General structure

```
string //The source of the grade update
```

XML-RPC (PHP structure)

```
[source] => string
```

REST (POST parameters)

```
source= string
```

`courseid` (Required)

id of course

General structure

```
int //id of course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

`component` (Required)

A component, for example mod_forum or mod_quiz

General structure

```
string //A component, for example mod_forum or mod_quiz
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

activityid (Required)

The activity ID

General structure

```
int //The activity ID
```

XML-RPC (PHP structure)

```
[activityid] => int
```

REST (POST parameters)

```
activityid= int
```

itemnumber (Required)

grade item ID number for modules that have multiple grades. Typically this is 0.

General structure

```
int //grade item ID number for modules that have multiple grades. Typically this is 0.
```

XML-RPC (PHP structure)

```
[itemnumber] => int
```

REST (POST parameters)

```
itemnumber= int
```

grades (Default to "Array ()")

Any student grades to alter

General structure

```
Default to "Array
(
)
" //Any student grades to alter
list of (
object {
studentid int //Student ID
grade double //Student grade
str_feedback string Optional //A string representation of the feedback from the grader
}
)
```



XML-RPC (PHP structure)

```
[grades] =>
  Array
    (
      [0] =>
        Array
          (
            [studentid] => int
            [grade] => double
            [str_feedback] => string
          )
    )
```

REST (POST parameters)

```
grades[0][studentid]= int
grades[0][grade]= double
grades[0][str_feedback]= string
```

`itemdetails` (Default to "Array ()")
Any grade item settings to alter

```
General structure

Default to "Array"
(
)
" //Any grade item settings to alter
object {
    itemname string  Optional //The grade item name
    idnumber int  Optional //Arbitrary ID provided by the module responsible for the grade item
    gradetype int  Optional //The type of grade (0 = none, 1 = value, 2 = scale, 3 = text)
    grademax double  Optional //Maximum grade allowed
    grademin double  Optional //Minimum grade allowed
    scaleid int  Optional //The ID of the custom scale being is used
    multfactor double  Optional //Multiply all grades by this number
    plusfactor double  Optional //Add this to all grades
    deleted int  Optional //True if the grade item should be deleted
    hidden int  Optional //True if the grade item is hidden
}
```

XML-RPC (PHP structure)

```
[itemdetails] =>
    Array
        (
            [itemname] => string
            [idnumber] => int
            [gradetype] => int
            [grademax] => double
            [grademin] => double
            [scaleid] => int
            [multfactor] => double
            [plusfactor] => double
            [deleted] => int
            [hidden] => int
        )
```

REST (POST parameters)

```
itemdetails[itemname]= string
itemdetails[idnumber]= int
itemdetails[gradetype]= int
itemdetails[grademax]= double
itemdetails[grademin]= double
itemdetails[scaleid]= int
itemdetails[multfactor]= double
itemdetails[plusfactor]= double
itemdetails[deleted]= int
itemdetails[hidden]= int
```

Response

A value like 0 => OK, 1 => FAILED as defined in lib/grade/constants.php

General structure

```
int  //A value like 0 => OK, 1 => FAILED
     as defined in lib/grade/constants.php
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_grading_get_definitions

Get grading definitions

Arguments

cmids (Required)

1 or more course module ids

General structure

```
//1 or more course module ids
list of (
int //course module id
)
```

XML-RPC (PHP structure)

```
[cmids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

cmids[0]= int

areaname (Required)

area name

General structure

string //area name

XML-RPC (PHP structure)

```
[areaname] => string
```

REST (POST parameters)

```
areaname= string
```

activeonly (Default to "0")

Only the active method

General structure

```
int Default to "0" //Only the active method
```

XML-RPC (PHP structure)

```
[activeonly] => int
```

REST (POST parameters)

```
activeonly= int
```

Response



```
General structure

object {
areas //list of grading areas
list of (
object {
cmid int //course module id
contextid int //context id
component string //component name
areaname string //area name
activemethod string Optional //active method
definitions //definitions
list of (
object {
id int Optional //definition id
method string //method
name string //name
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
status int //status
copiedfromid int Optional //copied from id
timecreated int //creation time
usercreated int //user who created definition
timemodified int //Last modified time
usermodified int //user who modified definition
timecopied int Optional //time copied
guide Optional //items
object {
guide_criteria list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
shortname string //description
descriptionmarkers string Optional //markers description
descriptionmarkersformat int Optional //descriptionmarkers format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
maxscore double //maximum score
})
)guide_comments Optional //comments
list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
})
)}
rubric Optional //items
object {
rubric_criteria Optional //definition details
list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
levels Optional //levels
list of (
object {
id int Optional //level id
score double Optional //score
definition string Optional //definition
definitionformat int Optional //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
})
)})
)})
})
})
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)}
```

XML-RPC (PHP structure)

```

Array
(
    [areas] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [cmid] => int
                            [contextid] => int
                            [component] => string
                            [areaname] => string
                            [activemethod] => string
                            [definitions] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [method] => string
                                                    [name] => string
                                                    [description] => string
                                                    [descriptionformat] => int
                                                    [status] => int
                                                    [copiedfromid] => int
                                                    [timecreated] => int
                                                    [usercreated] => int
                                                    [timemodified] => int
                                                    [usermodified] => int
                                                    [timecopied] => int
                                                    [guide] =>
                                                        Array
                                                            (
                                                                [guide_criteria] =>
                                                                    Array
                                                                        (
                                                                            [0] =>
                                                                                Array
                                                                                    (
                                                                                        [id] => int
                                                                                        [sortorder] => int
                                                                                        [description] => string
                                                                                        [descriptionformat] => int
                                                                                        [shortname] => string
                                                                                        [descriptionmarkers] => string
                                                                                        [descriptionmarkersformat] => int
                                                                                        [maxscore] => double
                                                                                    )
                                                                        )
                                                                )
                                                            )
                                                        )
                                                    [guide_comments] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                            [id] => int
                                                                            [sortorder] => int
                                                                            [description] => string
                                                                            [descriptionformat] => int
                                                                        )
                                                            )
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            [rubric] =>
                                Array
                                    (
                                        [rubric_criteria] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [

```





REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="areas">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="cmid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="contextid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="areaname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="activemethod">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="definitions">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="method">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="description">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="descriptionformat">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="status">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="copiedfromid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="usercreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="usermodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecopied">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="guide">
                                    <SINGLE>
                                        <KEY name="guide_criteria">
                                            <MULTIPLE>
                                                <SINGLE>
                                                    <KEY name="id">
                                                        <VALUE>int</VALUE>
                                                    </KEY>
                                                    <KEY name="sortorder">
                                                        <VALUE>int</VALUE>
                                                    </KEY>
                                                    <KEY name="description">
                                                        <VALUE>string</VALUE>
                                                    </KEY>
                                                    <KEY name="descriptionformat">
                                                        <VALUE>int</VALUE>
                                                    </KEY>
                                                    <KEY name="shortname">
                                                        <VALUE>string</VALUE>
                                                    </KEY>
                                                    <KEY name="descriptionmarkers">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionmarkersformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxscore">
    <VALUE>double</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="guide_comments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="sortorder">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
<KEY name="rubric">
    <SINGLE>
        <KEY name="rubric_criteria">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="levels">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="score">
                                    <VALUE>double</VALUE>
                                </KEY>
                                <KEY name="definition">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="definitionformat">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>

```



```

<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_grading_get_gradingform_instances

Get grading form instances

Arguments

definitionid (Required)

definition id

General structure

int //definition id

XML-RPC (PHP structure)

[definitionid] => int

REST (POST parameters)

definitionid= int

since (Default to "0")

submitted since

General structure

int Default to "0" //submitted since

XML-RPC (PHP structure)

```
[since] => int
```

REST (POST parameters)

```
since= int
```

Response**General structure**

```
object {
instances //List of grading instances
list of (
object {
id int //instance id
raterid int //rater id
itemid int //item id
rawgrade string Optional //raw grade
status int //status
feedback string Optional //feedback
feedbackformat int Optional //feedback format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
timemodified int //modified time
guide Optional //items
object {
criteria Optional //filling
list of (
object {
id int //filling id
criterionid int //criterion id
levelid int Optional //level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
score double //maximum score
})
})
rubric Optional //items
object {
criteria Optional //filling
list of (
object {
id int //filling id
criterionid int //criterion id
levelid int Optional //level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
})
})
}
)
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
})
```



XML-RPC (PHP structure)

```

Array
(
    [
        [instances] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [raterid] => int
                                [itemid] => int
                                [rawgrade] => string
                                [status] => int
                                [feedback] => string
                                [feedbackformat] => int
                                [timemodified] => int
                                [guide] =>
                                    Array
                                    (
                                        [
                                            [criteria] =>
                                                Array
                                                (
                                                    [
                                                        [0] =>
                                                            Array
                                                            (
                                                                [
                                                                    [id] => int
                                                                    [criterionid] => int
                                                                    [levelid] => int
                                                                    [remark] => string
                                                                    [remarkformat] => int
                                                                    [score] => double
                                                                )
                                                            )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [
            [rubric] =>
                Array
                (
                    [
                        [criteria] =>
                            Array
                            (
                                [
                                    [0] =>
                                        Array
                                        (
                                            [
                                                [id] => int
                                                [criterionid] => int
                                                [levelid] => int
                                                [remark] => string
                                                [remarkformat] => int
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
[warnings] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
        )
    )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="instances">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="raterid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="rawgrade">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="status">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="feedback">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="feedbackformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="guide">
            <SINGLE>
              <KEY name="criteria">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="id">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="criterionid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="levelid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="remark">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="remarkformat">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="score">
                      <VALUE>double</VALUE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </SINGLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
<KEY name="rubric">
  <SINGLE>
    <KEY name="criteria">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="criterionid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="levelid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="remark">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="remarkformat">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</KEY>
```



```

        </SINGLE>
    </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
    <SINGLE>
        <KEY name="item">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warningcode">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="message">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_grading_save_definitions

Save grading definitions



Arguments

areas (Required)

areas with definitions to save

```
//areas with definitions to save
list of (
object {
cmid int //course module id
contextid int //context id
component string //component name
areaname string //area name
activemethod string Optional //active method
definitions //definitions
list of (
object {
id int Optional //definition id
method string //method
name string //name
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
status int //status
copiedfromid int Optional //copied from id
timecreated int //creation time
usercreated int //user who created definition
timemodified int //last modified time
usermodified int //user who modified definition
timecopied int Optional //time copied
guide Optional //items
object {
guide_criteria list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
shortname string //description
descriptionmarkers string Optional //markers description
descriptionmarkersformat int Optional //descriptionmarkers format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
maxscore double //maximum score
}
)guide_comments Optional //comments
list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
))
rubric Optional //items
object {
rubric_criteria Optional //definition details
list of (
object {
id int Optional //criterion id
sortorder int Optional //sortorder
description string Optional //description
descriptionformat int Optional //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
levels Optional //levels
list of (
object {
id int Optional //level id
score double Optional //score
definition string Optional //definition
definitionformat int Optional //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
))
})
})
})
})
```

XML-RPC (PHP structure)

```
[areas] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [cmid] => int
                [contextid] => int
                [component] => string
                [areaname] => string
                [activemethod] => string
                [definitions] =>
                  Array
                    (
                      [
                        [0] =>
                          Array
                            (
                              [
                                [id] => int
                                [method] => string
                                [name] => string
                                [description] => string
                                [descriptionformat] => int
                                [status] => int
                                [copiedfromid] => int
                                [timecreated] => int
                                [usercreated] => int
                                [timemodified] => int
                                [usermodified] => int
                                [timecopied] => int
                                [guide] =>
                                  Array
                                    (
                                      [
                                        [guide_criteria] =>
                                          Array
                                            (
                                              [
                                                [0] =>
                                                  Array
                                                    (
                                                      [
                                                        [id] => int
                                                        [sortorder] => int
                                                        [description] => string
                                                        [descriptionformat] => int
                                                        [shortname] => string
                                                        [descriptionmarkers] => string
                                                        [descriptionmarkersformat] => int
                                                        [maxscore] => double
                                                      )
                                                    )
                                              )
                                            )
                                          )
                                        )
                                      )
                                    )
                                  )
                                )
                              )
                            )
                          )
                        )
                      )
                    )
                  )
                )
              )
            )
          )
        )
      )
    )
  )
)
[rubric] =>
  Array
    (
      [
        [rubric_criteria] =>
          Array
            (
              [
                [0] =>
                  Array
                    (
                      [
                        [id] => int
                        [sortorder] => int
                        [description] => string
                        [descriptionformat] => int
                      )
                    )
                  )
                )
              )
            )
          )
        )
      )
    )
  )
)

```

REST (POST parameters)

```
areas[0][cmd]= int
areas[0][contextid]= int
areas[0][component]= string
areas[0][areaname]= string
areas[0][activemethod]= string
areas[0][definitions][0][id]= int
areas[0][definitions][0][method]= string
areas[0][definitions][0][name]= string
areas[0][definitions][0][description]= string
areas[0][definitions][0][descriptionformat]= int
areas[0][definitions][0][status]= int
areas[0][definitions][0][copiedfromid]= int
areas[0][definitions][0][timecreated]= int
areas[0][definitions][0][usercreated]= int
areas[0][definitions][0][timemodified]= int
areas[0][definitions][0][usermodified]= int
areas[0][definitions][0][timecopied]= int
areas[0][definitions][0][guide][guide_criteria][0][id]= int
areas[0][definitions][0][guide][guide_criteria][0][sortorder]= int
areas[0][definitions][0][guide][guide_criteria][0][description]= string
areas[0][definitions][0][guide][guide_criteria][0][descriptionformat]= int
areas[0][definitions][0][guide][guide_criteria][0][shortname]= string
areas[0][definitions][0][guide][guide_criteria][0][descriptionmarkers]= string
areas[0][definitions][0][guide][guide_criteria][0][descriptionmarkersformat]= int
areas[0][definitions][0][guide][guide_criteria][0][maxscore]= double
areas[0][definitions][0][guide][guide_comments][0][id]= int
areas[0][definitions][0][guide][guide_comments][0][sortorder]= int
areas[0][definitions][0][guide][guide_comments][0][description]= string
areas[0][definitions][0][guide][guide_comments][0][descriptionformat]= int
areas[0][definitions][0][rubric][rubric_criteria][0][id]= int
areas[0][definitions][0][rubric][rubric_criteria][0][sortorder]= int
areas[0][definitions][0][rubric][rubric_criteria][0][description]= string
areas[0][definitions][0][rubric][rubric_criteria][0][descriptionformat]= int
areas[0][definitions][0][rubric][rubric_criteria][0][levels][0][id]= int
areas[0][definitions][0][rubric][rubric_criteria][0][levels][0][score]= double
areas[0][definitions][0][rubric][rubric_criteria][0][levels][0][definition]= string
areas[0][definitions][0][rubric][rubric_criteria][0][levels][0][definitionformat]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_add_group_members

Adds group members.

Arguments

members (Required)

General structure

```
list of (
object {
  groupid int //group record id
  userid int //user id
}
)
```

XML-RPC (PHP structure)

```
[members] =>
  Array
    (
      [0] =>
        Array
          (
            [groupid] => int
            [userid] => int
          )
    )
)
```

REST (POST parameters)

```
members[0][groupid]= int
members[0][userid]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_assign_grouping

Assing groups from groupings

Arguments

assignments (Required)

General structure

```
list of (
object {
groupingid int    //grouping record id
groupid int     //group record id
}
)
```

XML-RPC (PHP structure)

```
[assignments] =>
Array
(
[0] =>
Array
(
[groupingid] => int
[groupid] => int
)
)
```

REST (POST parameters)

```
assignments[0][groupingid]= int
assignments[0][groupid]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_create_groupings

Creates new groupings

Arguments

groupings (Required)

List of grouping object. A grouping has a courseid, a name and a description.

General structure

```
//List of grouping object. A grouping has a courseid, a name and a description.
list of (
object {
courseid int //id of course
name string //multilang compatible name, course unique
description string //grouping description text
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string Optional //id number
}
)
```

XML-RPC (PHP structure)

```
[groupings] =>
Array
(
[0] =>
Array
(
[courseid] => int
[name] => string
[description] => string
[descriptionformat] => int
[idnumber] => string
)
)
```

REST (POST parameters)

```
groupings[0][courseid]= int
groupings[0][name]= string
groupings[0][description]= string
groupings[0][descriptionformat]= int
groupings[0][idnumber]= string
```

Response

List of grouping object. A grouping has an id, a courseid, a name and a description.

General structure

```
//List of grouping object. A grouping has an id, a courseid, a name and a description.
list of (
object {
id int //grouping record id
courseid int //id of course
name string //multilang compatible name, course unique
description string //grouping description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string //id number
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
Array
(
[id] => int
[courseid] => int
[name] => string
[description] => string
[descriptionformat] => int
[idnumber] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_create_groups

Creates new groups.

**Arguments****groups** (Required)

List of group object. A group has a courseid, a name, a description and an enrolment key.

General structure

```
//List of group object. A group has a courseid, a name, a description and an enrolment key.
list of (
object {
courseid int //id of course
name string //multilang compatible name, course unique
description string //group description text
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string Optional //group enrol secret phrase
idnumber string Optional //id number
visibility int Default to "0" //group visibility mode. 0 = Visible to all. 1 = Visible to members. 2 = See own membership. 3 = Membership is hidden.
default: 0
participation int Default to "1" //activity participation enabled? Only for "all" and "members" visibility. Default true.
}
)
```

XML-RPC (PHP structure)

```
[groups] =>
Array
(
[0] =>
Array
(
[courseid] => int
[name] => string
[description] => string
[descriptionformat] => int
=enrolmentkey] => string
[idnumber] => string
[visibility] => int
[participation] => int
)
)
```

REST (POST parameters)

```
groups[0][courseid]= int
groups[0][name]= string
groups[0][description]= string
groups[0][descriptionformat]= int
groups[0][enrolmentkey]= string
groups[0][idnumber]= string
groups[0][visibility]= int
groups[0][participation]= int
```

Response

List of group object. A group has an id, a courseid, a name, a description and an enrolment key.

General structure

```
//List of group object. A group has an id, a courseid, a name, a description and an enrolment key.
list of (
object {
id int //group record id
courseid int //id of course
name string //multilang compatible name, course unique
description string //group description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string //group enrol secret phrase
idnumber string //id number
visibility int //group visibility mode. 0 = Visible to all. 1 = Visible to members. 2 = See own membership. 3 = Membership is hidden.
participation int //participation mode
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [id] => int
                    [courseid] => int
                    [name] => string
                    [description] => string
                    [descriptionformat] => int
                    [enrolmentkey] => string
                    [idnumber] => string
                    [visibility] => int
                    [participation] => int
                )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="enrolmentkey">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="visibility">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="participation">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_delete_group_members

Deletes group members.

Arguments

members (Required)

General structure

```
list of (
object {
  groupid int //group record id
  userid int //user id
}
)
```

XML-RPC (PHP structure)

```
[members] =>
  Array
    (
      [0] =>
        Array
          (
            [groupid] => int
            [userid] => int
          )
    )
)
```

REST (POST parameters)

```
members[0][groupid]= int
members[0][userid]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_delete_groupings

Deletes all specified groupings.

Arguments

groupingids (Required)

General structure

```
list of (
int   //grouping ID
)
```

XML-RPC (PHP structure)

```
[groupingids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
groupingids[0]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes



Callable from AJAX

No

core_group_delete_groups

Deletes all specified groups.

Arguments

groupids (Required)

General structure

```
list of (
int   //Group ID
)
```

XML-RPC (PHP structure)

```
[groupids] =>
    Array
    (
        [
            [0] => int
        )
    )
```

REST (POST parameters)

```
groupids[0]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_activity_allowed_groups

Gets a list of groups that the user is allowed to access within the specified activity.

Arguments

cmid (Required)

course module id

**General structure**

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

userid (Default to "0")

id of user, empty for current user

General structure

```
int Default to "0" //id of user, empty for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
groups list of (
object {
id int //group record id
name string //group name
description string //group description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string //id number
courseid int Optional //course id
}
)canaccessallgroups int Optional //Whether the user will be able to access all the activity groups.
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
[groups] =>
  Array
    (
      [0] =>
        Array
          (
            [id] => int
            [name] => string
            [description] => string
            [descriptionformat] => int
            [idnumber] => string
            [courseid] => int
          )
    )
  [canaccessallgroups] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groups">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="canaccessallgroups">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_activity_groupmode

Returns effective groupmode used in a given activity.

Arguments**cmid** (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

Response**General structure**

```
object {
  groupmode int //group mode:                                     0 for no groups, 1 for separate groups, 2 for visible groups
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [groupmode] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groupmode">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**core_group_get_course_groupings**

Returns all groupings in specified course.

Arguments**courseid** (Required)

id of course

General structure

int //id of course

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
list of (
object {
id int //grouping record id
courseid int //id of course
name string //multilang compatible name, course unique
description string //grouping description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string //id number
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[id] => int
[courseid] => int
[name] => string
[description] => string
[descriptionformat] => int
[idnumber] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_course_groups

Returns all groups in specified course.

Arguments**courseid** (Required)

id of course

General structure

```
int //id of course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
list of (
object {
id int //group record id
courseid int //id of course
name string //group name
description string //group description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string //group enrol secret phrase
idnumber string //id number
visibility int //group visibility mode. 0 = Visible to all. 1 = Visible to members. 2 = See own membership. 3 = Membership is hidden.
participation int //participation mode
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [id] => int
                    [courseid] => int
                    [name] => string
                    [description] => string
                    [descriptionformat] => int
                    [enrolmentkey] => string
                    [idnumber] => string
                    [visibility] => int
                    [participation] => int
                )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="enrolmentkey">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="visibility">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="participation">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_group_get_course_user_groups

Returns all groups in specified course for the specified user.

Arguments**courseid** (Default to "0")

Id of course (empty or 0 for all the courses where the user is enrolled).

General structure

```
int Default to "0" //Id of course (empty or 0 for all the courses where the user is enrolled).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

Id of user (empty or 0 for current user).

General structure

```
int Default to "0" //Id of user (empty or 0 for current user).
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

groupingid (Default to "0")

returns only groups in the specified grouping

**General structure**

```
int Default to "0" //returns only groups in the specified grouping
```

XML-RPC (PHP structure)

```
[groupingid] => int
```

REST (POST parameters)

```
groupingid= int
```

Response

General structure

```
object {
groups list of (
object {
id int    //group record id
name string  //group name
description string  //group description text
descriptionformat int   //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string  //id number
courseid int  Optional //course id
)
}warnings  Optional //List of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)}`
```

XML-RPC (PHP structure)

```
Array
(
    [groups] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [name] => string
                            [description] => string
                            [descriptionformat] => int
                            [idnumber] => string
                            [courseid] => int
                        )
                    )
            )
        )
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groups">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_group_members

Returns group members.

Arguments

groupids (Required)

General structure

```
list of (
int //Group ID
)
```

XML-RPC (PHP structure)

```
[groupids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
groupids[0]= int
```

Response

General structure

```
list of (
object {
groupid int //group record id
userids list of (
int //user id
)
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[groupid] => int
[userids] =>
Array
(
[0] => int
)
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="groupid">
<VALUE>int</VALUE>
</KEY>
<KEY name="userids">
<MULTIPLE>
<VALUE>int</VALUE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_groupings

Returns groupings details.

Arguments

groupingids (Required)

List of grouping id. A grouping id is an integer.

General structure

```
//List of grouping id. A grouping id is an integer.
list of (
int //grouping ID
)
```

XML-RPC (PHP structure)

```
[groupingids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
groupingids[0]= int
```

returnngroups (Default to "0")

return associated groups

General structure

```
int Default to "0" //return associated groups
```

XML-RPC (PHP structure)

```
[returnngroups] => int
```

REST (POST parameters)

```
returnngroups= int
```

Response

```
General structure

list of (
object {
id int    //grouping record id
courseid int   //id of course
name string   //multilang compatible name, course unique
description string //grouping description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string //id number
groups Optional //optional groups
list of (
object {
id int    //group record id
courseid int   //id of course
name string   //multilang compatible name, course unique
description string //group description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string //group enrol secret phrase
idnumber string //id number
})
)
})
```

XML-RPC (PHP structure)

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="idnumber">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="groups">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="id">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="courseid">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="name">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="description">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="descriptionformat">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="enrolmentkey">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="idnumber">
                            <VALUE>string</VALUE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_group_get_groups

Returns group details.

Arguments**groupids** (Required)

List of group id. A group id is an integer.

General structure

```
//List of group id. A group id is an integer.
list of (
int //Group ID
)
```

XML-RPC (PHP structure)

```
[groupids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

groupids[0]= int

Response**General structure**

```
list of (
object {
id int //group record id
courseid int //id of course
name string //group name
description string //group description text
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string //group enrol secret phrase
idnumber string //id number
visibility int //group visibility mode. 0 = Visible to all. 1 = Visible to members. 2 = See own membership. 3 = Membership is hidden.
participation int //participation mode
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[id] => int
[courseid] => int
[name] => string
[description] => string
[descriptionformat] => int
[enrolmentkey] => string
[idnumber] => string
[visibility] => int
[participation] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="courseid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="enrolmentkey">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="visibility">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="participation">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

core_group_unassign_grouping

Unassing groups from groupings

Arguments

unassignments (Required)

General structure

```
list of (
object {
groupingid int //grouping record id
groupid int //group record id
}
)
```

XML-RPC (PHP structure)

```
[unassignments] =>
Array
(
[0] =>
Array
(
[groupingid] => int
[groupid] => int
)
)
```

REST (POST parameters)

```
unassignments[0][groupingid]= int
unassignments[0][groupid]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

core_group_update_groupings

Updates existing groupings

Arguments**groupings (Required)**

List of grouping object. A grouping has a courseid, a name and a description.

General structure

```
//List of grouping object. A grouping has a courseid, a name and a description.
list of (
object {
id int //id of grouping
name string //multilang compatible name, course unique
description string //grouping description text
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
idnumber string Optional //id number
}
)
```

XML-RPC (PHP structure)

```
[groupings] =>
Array
(
[0] =>
Array
(
[id] => int
[name] => string
[description] => string
[descriptionformat] => int
[idnumber] => string
)
)
```

REST (POST parameters)

```
groupings[0][id]= int
groupings[0][name]= string
groupings[0][description]= string
groupings[0][descriptionformat]= int
groupings[0][idnumber]= string
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

core_group_update_groups

Updates existing groups.

Arguments

groups (Required)

List of group objects. A group is found by the id, then all other details provided will be updated.

General structure

```
//List of group objects. A group is found by the id, then all other details provided will be updated.
list of (
object {
id int //ID of the group
name string //multilang compatible name, course unique
description string Optional //group description text
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
enrolmentkey string Optional //group enrol secret phrase
idnumber string Optional //id number
visibility string Optional //group visibility mode. 0 = Visible to all. 1 = Visible to members. 2 = See own membership. 3 = Membership is hidden.
participation int Optional //activity participation enabled? Only for "all" and "members" visibility
}
)
```

XML-RPC (PHP structure)

```
[groups] =>
Array
(
    [
    [0] =>
        Array
        (
            [
            [id] => int
            [name] => string
            [description] => string
            [descriptionformat] => int
            [enrolmentkey] => string
            [idnumber] => string
            [visibility] => string
            [participation] => int
            )
        )
)
```

REST (POST parameters)

```
groups[0][id]= int
groups[0][name]= string
groups[0][description]= string
groups[0][descriptionformat]= int
groups[0][enrolmentkey]= string
groups[0][idnumber]= string
groups[0][visibility]= string
groups[0][participation]= int
```

Response



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_hsp_get_trusted_hsp_file

Get the H5P file cleaned for Mobile App.

Arguments**url** (Required)

H5P file url.

General structure

```
string //H5P file url.
```

XML-RPC (PHP structure)

```
[url] => string
```

REST (POST parameters)

```
url= string
```

frame (Default to "0")

The frame allow to show the bar options below the content

General structure

```
int Default to "0" //The frame allow to show the bar options below the content
```

XML-RPC (PHP structure)

```
[frame] => int
```

REST (POST parameters)

```
frame= int
```

export (Default to "0")

The export allow to download the package

General structure

```
int Default to "0" //The export allow to download the package
```

**XML-RPC (PHP structure)**

```
[export] => int
```

REST (POST parameters)

```
export= int
```

embed (Default to "0")

The embed allow to copy the code to your site

General structure

```
int Default to "0" //The embed allow to copy the code to your site
```

XML-RPC (PHP structure)

```
[embed] => int
```

REST (POST parameters)

```
embed= int
```

copyright (Default to "0")

The copyright option

General structure

```
int Default to "0" //The copyright option
```

XML-RPC (PHP structure)

```
[copyright] => int
```

REST (POST parameters)

```
copyright= int
```

Response**General structure**

```
object {
  files //H5P file trusted.
  list of (
    //File.
  object {
    filename string Optional //File name.
    filepath string Optional //File path.
    filesize int Optional //File size.
    fileurl string Optional //Downloadable file url.
    timemodified int Optional //Time modified.
    mimetype string Optional //File mime type.
    isexternalfile int Optional //Whether is an external file.
    repositorytype string Optional //The repository type for external files.
  }
  )warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```



XML-RPC (PHP structure)

```
Array
(
    [files] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [filename] => string
                            [filepath] => string
                            [filesize] => int
                            [fileurl] => string
                            [timemodified] => int
                            [mimetype] => string
                            [isexternalfile] => int
                            [repositorytype] => string
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="files">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="filename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filepath">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filesize">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fileurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mimetype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isexternalfile">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repositorytype">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_block_user

Blocks a user

Arguments

userid (Required)

The id of the user who is blocking

General structure

```
int //The id of the user who is blocking
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

blockeduserid (Required)

The id of the user being blocked

General structure

```
int //The id of the user being blocked
```

XML-RPC (PHP structure)

```
[blockeduserid] => int
```

REST (POST parameters)

```
blockeduserid= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
  //warning
object {
  item string Optional //item
  itemid int Optional //item id
  warningcode string //the warning code can be used by the client app to implement specific behaviour
  message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                ]
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_message_confirm_contact_request

Confirms a contact request

Arguments**userid** (Required)

The id of the user making the request

General structure

```
int //The id of the user making the request
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

requesteduserid (Required)

The id of the user being requested

General structure

```
int //The id of the user being requested
```

XML-RPC (PHP structure)

```
[requesteduserid] => int
```

REST (POST parameters)

```
requesteduserid= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

**XML-RPC (PHP structure)**

```
Array
(
    [0] =>
        Array
        (
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_create_contact_request

Creates a contact request

**Arguments****userid** (Required)

The id of the user making the request

General structure

```
int //The id of the user making the request
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

requesteduserid (Required)

The id of the user being requested

General structure

```
int //The id of the user being requested
```

XML-RPC (PHP structure)

```
[requesteduserid] => int
```

REST (POST parameters)

```
requesteduserid= int
```

Response

General structure

```
object {
    request Optional //request record
    object {
        id int //Message id
        userid int //User from id
        requesteduserid int //User to id
        timecreated int //Time created
    }
    warnings Optional //List of warnings
    list of (
        //warning
    )
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [request] =>
        Array
            (
                [id] => int
                [userid] => int
                [requesteduserid] => int
                [timecreated] => int
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="request">
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="requesteduserid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_data_for_messagearea_search_messages

Retrieve the template data for searching for messages

Arguments**userid** (Required)

The id of the user who is performing the search

General structure

```
int //The id of the user who is performing the search
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

search (Required)

The string being searched

General structure

```
string //The string being searched
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

**limitnum (Default to "0")**

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

Response

General structure

```
object {
    contacts list of (
        object {
            userid int //The user's id
            fullname string //The user's name
            profileimageurl string //User picture URL
            profileimageurlsmall string //Small user picture URL
            ismessaging int //If we are messaging the user
            sentfromcurrentuser int //Was the last message sent from the current user?
            lastmessage string //The user's last message
            lastmessagedate int Default to "null" //Timestamp for last message
            messageid int Default to "null" //The unique search message id
            showonlinestatus int //Show the user's online status?
            isonline int //The user's online status
            isread int //If the user has read the message
            isblocked int //If the user has been blocked
            unreadcount int Default to "null" //The number of unread messages in this conversation
            conversationid int Default to "null" //The id of the conversation
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [contacts] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [userid] => int
                            [fullname] => string
                            [profileimageurl] => string
                            [profileimageurlsmall] => string
                            [ismessaging] => int
                            [sentfromcurrentuser] => int
                            [lastmessage] => string
                            [lastmessagedate] => int
                            [messageid] => int
                            [showonlinestatus] => int
                            [isonline] => int
                            [isread] => int
                            [isblocked] => int
                            [unreadcount] => int
                            [conversationid] => int
                        )
                    )
            )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="contacts">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurlsmall">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="ismessaging">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sentfromcurrentuser">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lastmessage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="lastmessagedate">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="messageid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="showonlinestatus">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isonline">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isread">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isblocked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="unreadcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="conversationid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_decline_contact_request

Declines a contact request

Arguments**userid** (Required)

The id of the user making the request

General structure

```
int //The id of the user making the request
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

requesteduserid (Required)

The id of the user being requested

General structure

```
int //The id of the user being requested
```

XML-RPC (PHP structure)

```
[requesteduserid] => int
```

REST (POST parameters)

```
requesteduserid= int
```

**Response**

list of warnings

General structure

```
Optional //List of warnings
list of (
  //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_message_delete_contacts

Remove contacts from the contact list

Arguments**userids (Required)**

List of user IDs

General structure

```
//List of user IDs
list of (
int //User ID
)
```

XML-RPC (PHP structure)

```
[userids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
userids[0]= int
```

userid (Default to "0")

The id of the user we are deleting the contacts for, 0 for the current user

General structure

```
int Default to "0" //The id of the user we are deleting the contacts for, 0 for the
current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_delete_conversations_by_id

Deletes a list of conversations.

Arguments

userid (Required)

The user id of who we want to delete the conversation for

General structure

```
int //The user id of who we want to delete the conversation for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversationids (Required)

List of conversation IDs

General structure

```
//List of conversation IDs
list of (
int //The id of the conversation
)
```

XML-RPC (PHP structure)

```
[conversationids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
conversationids[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_message_delete_message

Deletes a message.

Arguments**messageid** (Required)

The message id

General structure

```
int //The message id
```

XML-RPC (PHP structure)

```
[messageid] => int
```

REST (POST parameters)

```
messageid= int
```

userid (Required)

The user id of who we want to delete the message for

General structure

```
int //The user id of who we want to delete the message for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

read (Default to "1")

If is a message read

General structure

```
int Default to "1" //If is a message read
```

XML-RPC (PHP structure)

```
[read] => int
```

REST (POST parameters)

```
read= int
```

**Response****General structure**

```
object {
  status int //True if the message was deleted, false otherwise
  warnings Optional //List of warnings
  list of (
    //Warning
  object {
    item string Optional //Item
    itemid int Optional //Item id
    warningcode string //The warning code can be used by the client app to implement specific behaviour
    message string //Untranslated English message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
)
}

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_message_delete_message_for_all_users`

Deletes a message for all users.

Arguments

messageid (Required)

The message id

General structure

```
int //The message id
```

XML-RPC (PHP structure)

```
[messageid] => int
```

REST (POST parameters)

```
messageid= int
```

userid (Required)

The user id of who we want to delete the message for all users

General structure

```
int //The user id of who we want to delete the message for all users
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_blocked_users

Retrieve a list of users blocked

**Arguments****userid** (Required)

the user whose blocked users we want to retrieve

General structure

```
int //the user whose blocked users we want to retrieve
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

General structure

```
object {
    users //List of blocked users
    list of (
        object {
            id int //User ID
            fullname string //User full name
            profileimageurl string Optional //User picture URL
        }
    )warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [
        [users] =>
            Array
                (
                    [
                        [0] =>
                            Array
                                (
                                    [
                                        [id] => int
                                        [fullname] => string
                                        [profileimageurl] => string
                                    )
                                )
                )
        [warnings] =>
            Array
                (
                    [
                        [0] =>
                            Array
                                (
                                    [
                                        [item] => string
                                        [itemid] => int
                                        [warningcode] => string
                                        [message] => string
                                    )
                                )
                )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurl">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_message_get_contact_requests

Returns contact requests for a user

Arguments**userid** (Required)

The id of the user we want the requests for

General structure

```
int //The id of the user we want the requests for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

**Response**

General structure

```
list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The Link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
shownonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
}
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
}
)
)}
```



XML-RPC (PHP structure)



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isonline">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showonlinestatus">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isdeleted">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessageevenifblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="requirescontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contactrequests">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="requesteduserid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="conversations">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

</RESPONSE>

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversation

Retrieve a conversation for a user

Arguments**userid** (Required)

The id of the user who we are viewing conversations for

General structure

```
int //The id of the user who we are viewing conversations for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

**conversationid** (Required)

The id of the conversation to fetch

General structure

```
int //The id of the conversation to fetch
```

XML-RPC (PHP structure)

```
[conversationid] => int
```

REST (POST parameters)

```
conversationid= int
```

includecontactrequests (Required)

Include contact requests in the members

General structure

```
int //Include contact requests in the members
```

XML-RPC (PHP structure)

```
[includecontactrequests] => int
```

REST (POST parameters)

```
includecontactrequests= int
```

includeprivacyinfo (Required)

Include privacy info in the members

General structure

```
int //Include privacy info in the members
```

XML-RPC (PHP structure)

```
[includeprivacyinfo] => int
```

REST (POST parameters)

```
includeprivacyinfo= int
```

memberlimit (Default to "0")

Limit for number of members

General structure

```
int Default to "0" //Limit for number of members
```

**XML-RPC (PHP structure)**

```
[memberlimit] => int
```

REST (POST parameters)

```
memberlimit= int
```

memberoffset (Default to "0")

Offset for member list

General structure

```
int Default to "0" //Offset for member list
```

XML-RPC (PHP structure)

```
[memberoffset] => int
```

REST (POST parameters)

```
memberoffset= int
```

messagelimit (Default to "100")

Limit for number of messages

General structure

```
int Default to "100" //Limit for number of messages
```

XML-RPC (PHP structure)

```
[messagelimit] => int
```

REST (POST parameters)

```
messagelimit= int
```

messageoffset (Default to "0")

Offset for messages list

General structure

```
int Default to "0" //Offset for messages list
```

XML-RPC (PHP structure)

```
[messageoffset] => int
```

REST (POST parameters)

```
messageoffset= int
```

newestmessagesfirst (Default to "1")

Order messages by newest first

General structure

```
int Default to "1" //Order messages by newest first
```

XML-RPC (PHP structure)

```
[newestmessagesfirst] => int
```

REST (POST parameters)

```
newestmessagesfirst= int
```

**Response**

General structure

```

object {
id int //The conversation id
name string Default to "null" //The conversation name, if set
subname string Default to "null" //A subtitle for the conversation name, if set
imageurl string Default to "null" //A link to the conversation picture, if set
type int //The type of the conversation (1=individual,2=group,3=self)
membercount int //Total number of conversation members
ismuted int //If the user muted this conversation
isfavourite int //If the user marked this conversation as a favourite
isread int //If the user has read all messages in the conversation
unreadcount int Default to "null" //The number of unread messages in this conversation
members list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //Information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)}
)messages list of (
object {
id int //The id of the message
useridfrom int //The id of the user who sent the message
text string //The text of the message
timecreated int //The timecreated timestamp for the message
})
)candeletemessagesforallusers int Default to "" //If the user can delete messages in the conversation for all users
}

```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [name] => string
    [subname] => string
    [imageurl] => string
    [type] => int
    [membercount] => int
    [ismuted] => int
    [isfavourite] => int
    [isread] => int
    [unreadcount] => int
    [members] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [fullname] => string
                            [profileurl] => string
                            [profileimageurl] => string
                            [profileimageurlsmall] => string
                            [isonline] => int
                            [showonlinestatus] => int
                            [isblocked] => int
                            [iscontact] => int
                            [isdeleted] => int
                            [canmessageevenifblocked] => int
                            [canmessage] => int
                            [requirescontact] => int
                            [contactrequests] =>
                                Array
                                (
                                    [
                                        [0] =>
                                            Array
                                            (
                                                [
                                                    [id] => int
                                                    [userid] => int
                                                    [requesteduserid] => int
                                                    [timecreated] => int
                                                )
                                            )
                                    )
                                )
                            [conversations] =>
                                Array
                                (
                                    [
                                        [0] =>
                                            Array
                                            (
                                                [
                                                    [id] => int
                                                    [type] => int
                                                    [name] => string
                                                    [timecreated] => int
                                                )
                                            )
                                    )
                                )
                            )
                        )
                    )
    )
)
[messages] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [
                            [id] => int
                            [useridfrom] => int
                            [text] => string
                            [timecreated] => int
                        )
                    )
                )
        )
    )
)
[candeletemessagesforallusers] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="subname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="imageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="type">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="membercount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ismuted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isfavourite">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="unreadcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="members">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="isonline">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showonlinestatus">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="iscontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isdeleted">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessageevenifblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessage">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requirescontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="contactrequests">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
```



```

<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requesteduserid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="conversations">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="candeletemessagesforallusers">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversation_between_users

Retrieve a conversation for a user between another user

Arguments**userid** (Required)

The id of the user who we are viewing conversations for

General structure

```
int //The id of the user who we are viewing conversations for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

otheruserid (Required)

The other user id

General structure

```
int //The other user id
```

XML-RPC (PHP structure)

```
[otheruserid] => int
```

REST (POST parameters)

```
otheruserid= int
```

includecontactrequests (Required)

Include contact requests in the members

General structure

```
int //Include contact requests in the members
```

XML-RPC (PHP structure)

```
[includecontactrequests] => int
```

REST (POST parameters)

```
includecontactrequests= int
```

includeprivacyinfo (Required)

Include privacy info in the members

General structure

```
int //Include privacy info in the members
```

XML-RPC (PHP structure)

```
[includeprivacyinfo] => int
```

REST (POST parameters)

```
includeprivacyinfo= int
```

memberlimit (Default to "0")

Limit for number of members

General structure

```
int Default to "0" //Limit for number of members
```

XML-RPC (PHP structure)

```
[memberlimit] => int
```

REST (POST parameters)

```
memberlimit= int
```

memberoffset (Default to "0")

Offset for member list

General structure

```
int Default to "0" //Offset for member list
```

XML-RPC (PHP structure)

```
[memberoffset] => int
```

REST (POST parameters)

```
memberoffset= int
```

messagelimit (Default to "100")

Limit for number of messages

General structure

```
int Default to "100" //Limit for number of messages
```

XML-RPC (PHP structure)

```
[messagelimit] => int
```

REST (POST parameters)

```
messagelimit= int
```

messageoffset (Default to "0")

Offset for messages list

General structure

```
int Default to "0" //Offset for messages list
```

XML-RPC (PHP structure)

```
[messageoffset] => int
```

REST (POST parameters)

```
messageoffset= int
```

newestmessagesfirst (Default to "1")

Order messages by newest first

General structure

```
int Default to "1" //Order messages by newest first
```

XML-RPC (PHP structure)

```
[newestmessagesfirst] => int
```

REST (POST parameters)

```
newestmessagesfirst= int
```

Response

General structure

```

object {
id int //The conversation id
name string Default to "null" //The conversation name, if set
subname string Default to "null" //A subtitle for the conversation name, if set
imageurl string Default to "null" //A link to the conversation picture, if set
type int //The type of the conversation (1=individual,2=group,3=self)
membercount int //Total number of conversation members
ismuted int //If the user muted this conversation
isfavourite int //If the user marked this conversation as a favourite
isread int //If the user has read all messages in the conversation
unreadcount int Default to "null" //The number of unread messages in this conversation
members list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)}
)messages list of (
object {
id int //The id of the message
useridfrom int //The id of the user who sent the message
text string //The text of the message
timecreated int //The timecreated timestamp for the message
})
)candeletemessagesforallusers int Default to "" //If the user can delete messages in the conversation for all users
}

```

XML-RPC (PHP structure)



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="subname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="imageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="type">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="membercount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ismuted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isfavourite">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="unreadcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="members">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="isonline">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showonlinestatus">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="iscontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isdeleted">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessageevenifblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessage">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requirescontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="contactrequests">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
```



```

<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requesteduserid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="conversations">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="candeletemessagesforallusers">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversation_counts

Retrieve a list of conversation counts, indexed by type.

Arguments**userid** (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
    favourites int //Total number of favourite conversations
    types object {
        1 int //Total number of individual conversations
        2 int //Total number of group conversations
        3 int //Total number of self conversations
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [favourites] => int
    [types] =>
        Array
        (
            [1] => int
            [2] => int
            [3] => int
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="favourites">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="types">
            <SINGLE>
                <KEY name="1">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="2">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="3">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversation_members

Retrieve a list of members in a conversation

**Arguments****userid** (Required)

The id of the user we are performing this action on behalf of

General structure

```
int //The id of the user we are performing this action on behalf of
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversationid (Required)

The id of the conversation

General structure

```
int //The id of the conversation
```

XML-RPC (PHP structure)

```
[conversationid] => int
```

REST (POST parameters)

```
conversationid= int
```

includecontactrequests (Default to "")

Do we want to include contact requests?

General structure

```
int Default to "" //Do we want to include contact requests?
```

XML-RPC (PHP structure)

```
[includecontactrequests] => int
```

REST (POST parameters)

```
includecontactrequests= int
```

includeprivacyinfo (Default to "")

Do we want to include privacy info?

General structure

```
int Default to "" //Do we want to include privacy info?
```

XML-RPC (PHP structure)

```
[includeprivacyinfo] => int
```

REST (POST parameters)

```
includeprivacyinfo= int
```



limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

Response**General structure**

```
list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The Link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
shownlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)
)
```



XML-RPC (PHP structure)

```
    )
  [0] =>
    Array
      (
        [
          [id] => int
          [fullname] => string
          [profileurl] => string
          [profileimageurl] => string
          [profileimageurlsmall] => string
          [isonline] => int
          [showonlinestatus] => int
          [isblocked] => int
          [iscontact] => int
          [isdeleted] => int
          [canmessageevenifblocked] => int
          [canmessage] => int
          [requirescontact] => int
          [contactrequests] =>

          Array
            (
              [
                [0] =>
                  Array
                    (
                      [
                        [id] => int
                        [userid] => int
                        [requesteduserid] => int
                        [timecreated] => int
                      )
                    )
              )
            )
          )
        )
      )
    )
  )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isonline">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showonlinestatus">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isdeleted">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessageevenifblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="requirescontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contactrequests">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="requesteduserid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="conversations">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

</RESPONSE>

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversation_messages

Retrieve the conversation messages and relevant member information

Arguments**currentuserid** (Required)

The current user's id

General structure

int //The current user's id

XML-RPC (PHP structure)

[currentuserid] => int

REST (POST parameters)

currentuserid= int

**convid** (Required)

The conversation id

General structure

int //The conversation id

XML-RPC (PHP structure)

[convid] => int

REST (POST parameters)

convid= int

limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

newest (Default to "")

Newest first?

General structure

```
int Default to "" //Newest first?
```

XML-RPC (PHP structure)

```
[newest] => int
```

REST (POST parameters)

```
newest= int
```



timefrom (Default to "0")

The timestamp from which the messages were created

General structure

```
int Default to "0" //The timestamp from which the messages were created
```

XML-RPC (PHP structure)

```
[timefrom] => int
```

REST (POST parameters)

```
timefrom= int
```

Response

```
General structure

object {
id int //The conversation id
members list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The Link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)
)messages list of (
object {
id int //The id of the message
useridfrom int //The id of the user who sent the message
text string //The text of the message
timecreated int //The timecreated timestamp for the message
})
)
```



XML-RPC (PHP structure)

```

Array
(
    [id] => int
    [members] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [profileurl] => string
                    [profileimageurl] => string
                    [profileimageurlsmall] => string
                    [isonline] => int
                    [showonlinestatus] => int
                    [isblocked] => int
                    [iscontact] => int
                    [isdeleted] => int
                    [canmessageevenifblocked] => int
                    [canmessage] => int
                    [requirescontact] => int
                    [contactrequests] =>
                        Array
                        (
                            [0] =>
                                Array
                                (
                                    [id] => int
                                    [userid] => int
                                    [requesteduserid] => int
                                    [timecreated] => int
                                )
                        )
                )
            [conversations] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [id] => int
                            [type] => int
                            [name] => string
                            [timecreated] => int
                        )
                )
        )
    [messages] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [useridfrom] => int
                    [text] => string
                    [timecreated] => int
                )
        )
)

```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="members">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="isonline">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showonlinestatus">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="iscontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isdeleted">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessageevenifblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessage">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requirescontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="contactrequests">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="requesteduserid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="timecreated">
                  <VALUE>int</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
          <KEY name="conversations">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="type">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="timecreated">
                  <VALUE>int</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </MULTIPLE>
      </SINGLE>
    </MULTIPLE>
  </SINGLE>
</RESPONSE>
```



```

        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_conversations



Retrieve a list of conversations for a user

Arguments

userid (Required)

The id of the user who we are viewing conversations for

General structure

```
int //The id of the user who we are viewing conversations for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

limitfrom (Default to "0")

The offset to start at

General structure

```
int Default to "0" //The offset to start at
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number of conversations to this

General structure

```
int Default to "0" //Limit number of conversations to this
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

type (Default to "null")

Filter by type

General structure

```
int Default to "null" //Filter by type
```

**XML-RPC (PHP structure)**

```
[type] => int
```

REST (POST parameters)

```
type= int
```

favourites (Default to "null")

Whether to restrict the results to contain NO favourite conversations (false), ONLY favourite conversation (true), or ignore any restriction altogether (null)

General structure

```
int Default to "null" //Whether to restrict the results to contain NO favourite
conversations (false), ONLY favourite conversation (true), or ignore any restriction altogether (null)
```

XML-RPC (PHP structure)

```
[favourites] => int
```

REST (POST parameters)`favourites= int``mergeself (Default to "")`

Whether to include self-conversations (true) or ONLY private conversations (false) when private conversations are requested.

General structure

```
int Default to "" //Whether to include self-conversations (true) or ONLY private  
//conversations (false) when private conversations are requested.
```

XML-RPC (PHP structure)`[mergeself] => int`**REST (POST parameters)**`mergeself= int`**Response**

General structure

```
object {
    conversations list of (
        object {
            id int //The conversation id
            name string Default to "null" //The conversation name, if set
            subtitle string Default to "null" //A subtitle for the conversation name, if set
            imageurl string Default to "null" //A link to the conversation picture, if set
            type int //The type of the conversation (1=individual,2=group,3=self)
            membercount int //Total number of conversation members
            muted int //If the user muted this conversation
            isfavourite int //If the user marked this conversation as a favourite
            isread int //If the user has read all messages in the conversation
            unreadcount int Default to "null" //The number of unread messages in this conversation
            members list of (
                object {
                    id int //The user id
                    fullname string //The user's name
                    profileurl string //The link to the user's profile page
                    profileimageurl string //User picture URL
                    profileimageurlsmall string //Small user picture URL
                    isonline int //The user's online status
                    showonlinestatus int //Show the user's online status?
                    isblocked int //If the user has been blocked
                    iscontact int //Is the user a contact?
                    isdeleted int //Is the user deleted?
                    canmessageevenifblocked int //If the user can still message even if they get blocked
                    canmessage int //If the user can be messaged
                    requirescontact int //If the user requires to be contacts
                    contactrequests Optional //The contact requests
                }
            )
        }
    )
}

object {
    id int //The id of the contact request
    userid int //The id of the user who created the contact request
    requesteduserid int //The id of the user confirming the request
    timecreated int //The timecreated timestamp for the contact request
}

conversations Optional //Conversations between users
list of (
    Optional //Information about conversation
    object {
        id int //Conversations id
        type int //Conversation type: private or public
        name string //Multilang compatible conversation name2
        timecreated int //The timecreated timestamp for the conversation
    }
)
}

messages list of (
    object {
        id int //The id of the message
        useridfrom int //The id of the user who sent the message
        text string //The text of the message
        timecreated int //The timecreated timestamp for the message
    }
)
}

can deletemessagesforallusers int Default to "" //If the user can delete messages in the conversation for
}
```

XML-RPC (PHP structure)

```

Array
(
    [conversations] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [subname] => string
                            [imageurl] => string
                            [type] => int
                            [membercount] => int
                            [ismuted] => int
                            [isfavourite] => int
                            [isread] => int
                            [unreadcount] => int
                            [members] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [fullname] => string
                                                    [profileurl] => string
                                                    [profileimageurl] => string
                                                    [profileimageurlsmall] => string
                                                    [isonline] => int
                                                    [showonlinestatus] => int
                                                    [isblocked] => int
                                                    [iscontact] => int
                                                    [isdeleted] => int
                                                    [canmessageevenifblocked] => int
                                                    [canmessage] => int
                                                    [requirescontact] => int
                                                    [contactrequests] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                            [id] => int
                                                                            [userid] => int
                                                                            [requesteduserid] => int
                                                                            [timecreated] => int
                                                                        )
                                                                )
                                                        )
                                                    [conversations] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                            [id] => int
                                                                            [type] => int
                                                                            [name] => string
                                                                            [timecreated] => int
                                                                        )
                                                                )
                                                            )
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
)
[messages] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [useridfrom] => int
                        [text] => string
                        [timecreated] => int
                    )
            )
        )
[candeletemessagesforallusers] => int
)

```





^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="conversations">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="subname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="imageurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="type">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="membercount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="ismuted">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isfavourite">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isread">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="unreadcount">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="members">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fullname">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="profileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="profileimageurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="profileimageurlsmall">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isonline">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="showonlinestatus">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="isblocked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="iscontact">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="isdeleted">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="canmessageevenifblocked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="canmessage">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="requirescontact">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="contactrequests">
                                    <MULTIPLE>
                                        <SINGLE>
```



```

<KEY name="id">
    <VALUE>int</VALUE>
</KEY>
<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requesteduserid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="conversations">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="candeletemessagesforallusers">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_member_info

Retrieve a user message profiles

Arguments

referenceuserid (Required)

id of the user

General structure

```
int //id of the user
```

XML-RPC (PHP structure)

```
[referenceuserid] => int
```

REST (POST parameters)

```
referenceuserid= int
```

userids (Required)

General structure

```
list of (
int //id of members to get
)
```

XML-RPC (PHP structure)

```
[userids] =>
  Array
  (
    [0] => int
  )

```



REST (POST parameters)

```
userids[0]= int
```

includecontactrequests (Default to "")

include contact requests in response

General structure

```
int Default to "" //include contact requests in response
```

XML-RPC (PHP structure)

```
[includecontactrequests] => int
```

REST (POST parameters)

```
includecontactrequests= int
```

`includeprivacyinfo` (Default to "")
include privacy info in response

General structure

```
int Default to "" //include privacy info in response
```

XML-RPC (PHP structure)

```
[includeprivacyinfo] => int
```

REST (POST parameters)

```
includeprivacyinfo= int
```

Response**General structure**

```
list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)}
```



XML-RPC (PHP structure)



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isonline">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showonlinestatus">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isdeleted">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessageevenifblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="requirescontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contactrequests">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="requesteduserid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="conversations">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

</RESPONSE>

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_message_get_message_processor`

Get a message processor

Arguments**userid** (Required)

id of the user, 0 for current user

General structure

```
int //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

**name** (Required)

The name of the message processor

General structure

```
string //The name of the message processor
```

XML-RPC (PHP structure)

```
[name] => string
```

REST (POST parameters)

```
name= string
```

Response**General structure**

```
object {
    systemconfigured int //Site configuration status
    userconfigured int //The user configuration status
}
```

XML-RPC (PHP structure)

```
Array
(
    [systemconfigured] => int
    [userconfigured] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="systemconfigured">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="userconfigured">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

Yes

core_message_get_messages

Retrieve a list of messages sent and received by a user (conversations, notifications or both)

Arguments**useridto** (Required)

the user id who received the message, 0 for any user

General structure

```
int //the user id who received the message, 0 for any user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```

useridfrom (Default to "0")

the user id who send the message, 0 for any user. -10 or -20 for no-reply or support user

General structure

```
int Default to "0" //the user id who send the message, 0 for any user. -10 or -20 for no-reply or support user
```

XML-RPC (PHP structure)

```
[useridfrom] => int
```

REST (POST parameters)

```
useridfrom= int
```

type (Default to "both")

type of message to return, expected values are: notifications, conversations and both

General structure

```
string Default to "both" //type of message to return, expected values are: notifications, conversations and both
```

XML-RPC (PHP structure)

```
[type] => string
```

REST (POST parameters)

```
type= string
```

read (Default to "1")

1 for getting read messages, 0 for unread, 2 for both

General structure

```
int Default to "1" //1 for getting read messages, 0 for unread, 2 for both
```

XML-RPC (PHP structure)

```
[read] => int
```

REST (POST parameters)

```
read= int
```

newestfirst (Default to "1")

true for ordering by newest first, false for oldest first

General structure

```
int Default to "1" //true for ordering by newest first, false for oldest first
```

XML-RPC (PHP structure)

```
[newestfirst] => int
```

REST (POST parameters)

```
newestfirst= int
```

limitfrom (Default to "0")

limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

Response



General structure

```
object {
  messages list of (
    //message
  object {
    id int //Message id
    useridfrom int //User from id
    useridto int //User to id
    subject string //The message subject
    text string //The message text formated
    fullmessage string //The message
    fullmessageformat int //fullmessage format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    fullmessagehtml string //The message in html
    smallmessage string //The shorten message
    notification int //Is a notification?
    contexturl string //Context URL
    contexturlname string //Context URL Link name
    timecreated int //Time created
    timeread int //Time read
    userfullname string //User to full name
    userfromfullname string //User from full name
    component string Optional //The component that generated the notification
    eventtype string Optional //The type of notification
    customdata string Optional //Custom data to be passed to the message processor.
      The data here is serialised using json_encode().
    iconurl string Optional //URL for icon, only for notifications.
  }
  )warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [messages] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [useridfrom] => int
                                [useridto] => int
                                [subject] => string
                                [text] => string
                                [fullmessage] => string
                                [fullmessageformat] => int
                                [fullmessagehtml] => string
                                [smallmessage] => string
                                [notification] => int
                                [contexturl] => string
                                [contexturlname] => string
                                [timecreated] => int
                                [timeread] => int
                                [userfullname] => string
                                [userfromfullname] => string
                                [component] => string
                                [eventtype] => string
                                [customdata] => string
                                [iconurl] => string
                            )
                        )
                ]
            )
        [
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    ]
                )
            )
        )
    )
)
```



REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="messages">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="useridfrom">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="useridto">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="subject">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="text">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="fullmessage">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="fullmessageformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullmessagehtml">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="smallmessage">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="notification">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="contexturl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="contexturlname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeread">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usertofullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="userfromfullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="component">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="eventtype">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="customdata">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="iconurl">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </RESPONSE>

```

```

<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_message_get_received_contact_requests_count`

Gets the number of received contact requests

Arguments

`userid` (Required)

The id of the user we want to return the number of received contact requests for

General structure

```
int //The id of the user we want to return the number of received contact requests for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

The number of received contact requests

General structure

```
int //The number of received contact requests
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_self_conversation

Retrieve a self-conversation for a user

Arguments**userid** (Required)

The id of the user who we are viewing self-conversations for

General structure

```
int //The id of the user who we are viewing self-conversations for
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

messagelimit (Default to "100")

Limit for number of messages

General structure

```
int Default to "100" //Limit for number of messages
```

XML-RPC (PHP structure)

```
[messagelimit] => int
```

REST (POST parameters)

```
messagelimit= int
```

messageoffset (Default to "0")

Offset for messages list

General structure

```
int Default to "0" //Offset for messages list
```

XML-RPC (PHP structure)

```
[messageoffset] => int
```

REST (POST parameters)

```
messageoffset= int
```

newestmessagesfirst (Default to "1")

Order messages by newest first

General structure

```
int Default to "1" //Order messages by newest first
```

XML-RPC (PHP structure)

```
[newestmessagesfirst] => int
```

REST (POST parameters)

```
newestmessagesfirst= int
```

Response



General structure

```

object {
id int //The conversation id
name string Default to "null" //The conversation name, if set
subname string Default to "null" //A subtitle for the conversation name, if set
imageurl string Default to "null" //A link to the conversation picture, if set
type int //The type of the conversation (1=individual,2=group,3=self)
membercount int //Total number of conversation members
ismuted int //If the user muted this conversation
isfavourite int //If the user marked this conversation as a favourite
isread int //If the user has read all messages in the conversation
unreadcount int Default to "null" //The number of unread messages in this conversation
members list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //Information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)}
)messages list of (
object {
id int //The id of the message
useridfrom int //The id of the user who sent the message
text string //The text of the message
timecreated int //The timecreated timestamp for the message
})
)candeletemessagesforallusers int Default to "" //If the user can delete messages in the conversation for all users
}

```

XML-RPC (PHP structure)

```

Array
(
    [id] => int
    [name] => string
    [subname] => string
    [imageurl] => string
    [type] => int
    [membercount] => int
    [ismuted] => int
    [isfavourite] => int
    [isread] => int
    [unreadcount] => int
    [members] =>
        Array
            (
                [
                    [0] =>
                        Array
                            (
                                [id] => int
                                [fullname] => string
                                [profileurl] => string
                                [profileimageurl] => string
                                [profileimageurlsmall] => string
                                [isonline] => int
                                [showonlinestatus] => int
                                [isblocked] => int
                                [iscontact] => int
                                [isdeleted] => int
                                [canmessageevenifblocked] => int
                                [canmessage] => int
                                [requirescontact] => int
                                [contactrequests] =>
                                    Array
                                        (
                                            [
                                                [0] =>
                                                    Array
                                                        (
                                                            [id] => int
                                                            [userid] => int
                                                            [requesteduserid] => int
                                                            [timecreated] => int
                                                        )
                                            )
                                        )
                                [conversations] =>
                                    Array
                                        (
                                            [
                                                [0] =>
                                                    Array
                                                        (
                                                            [
                                                                [id] => int
                                                                [type] => int
                                                                [name] => string
                                                                [timecreated] => int
                                                            )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                )
            )
)
[messages] =>
    Array
        (
            [
                [0] =>
                    Array
                        (
                            [
                                [id] => int
                                [useridfrom] => int
                                [text] => string
                                [timecreated] => int
                            )
                        )
            )
        )
[candeletemessagesforallusers] => int
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="subname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="imageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="type">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="membercount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ismuted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isfavourite">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isread">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="unreadcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="members">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurl">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="isonline">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="showonlinestatus">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="iscontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="isdeleted">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessageevenifblocked">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmessage">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requirescontact">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="contactrequests">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
```



```

<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requesteduserid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="conversations">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="candeletemessagesforallusers">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_unread_conversation_counts

Retrieve a list of unread conversation counts, indexed by type.

Arguments

userid (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
    favourites int //Total number of unread favourite conversations
    types object {
        1 int //Total number of unread individual conversations
        2 int //Total number of unread group conversations
        3 int //Total number of unread self conversations
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [favourites] => int
    [types] =>
        Array
        (
            [1] => int
            [2] => int
            [3] => int
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="favourites">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="types">
      <SINGLE>
        <KEY name="1">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="2">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="3">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_unread_conversations_count

Retrieve the count of unread conversations for a given user

**Arguments****useridto (Required)**

the user id who received the message, 0 for any user

General structure

```
int //the user id who received the message, 0 for any user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```

Response

The count of unread messages for the user

General structure

```
int //The count of unread messages for the user
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_message_get_unread_notification_count`

Get number of unread notifications.



Arguments

`useridto` (Required)

user id who received the notification, 0 for any user

General structure

```
int //user id who received the notification, 0 for any user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```

Response

The count of unread notifications.

General structure

```
int //The count of unread notifications.
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_message_get_user_contacts

Retrieve the contact list



Arguments

userid (Required)

The id of the user who we retrieving the contacts for

General structure

```
int //The id of the user who we retrieving the contacts for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

Response**General structure**

```
list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
showonlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //Information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)}
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [fullname] => string
                [profileurl] => string
                [profileimageurl] => string
                [profileimageurlsmall] => string
                [isonline] => int
                [showonlinestatus] => int
                [isblocked] => int
                [iscontact] => int
                [isdeleted] => int
                [canmessageevenifblocked] => int
                [canmessage] => int
                [requirescontact] => int
                [contactrequests] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [id] => int
                                        [userid] => int
                                        [requesteduserid] => int
                                        [timecreated] => int
                                    )
                        )
                )
            )
        [conversations] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [id] => int
                                [type] => int
                                [name] => string
                                [timecreated] => int
                            )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isonline">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="showonlinestatus">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="iscontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isdeleted">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessageevenifblocked">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="canmessage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="requirescontact">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="contactrequests">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="requesteduserid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="conversations">
        <MULTIPLE>
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="type">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

</RESPONSE>

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_user_message_preferences

Get the message preferences for a given user.

Arguments

userid (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```



Response

General structure

```

object {
preferences object {
userid int   //User id
disableall int  //Whether all the preferences are disabled
processors  //Config form values
list of (
object {
displayname string  //Display name
name string  //Processor name
hassettings int  //Whether has settings
contextid int  //Context id
userconfigured int  //Whether is configured by the user
})
components  //Available components
list of (
object {
displayname string  //Display name
notifications  //List of notifications for the component
list of (
object {
displayname string  //Display name
preferencekey string  //Preference key
processors  //Processors values for this notification
list of (
object {
displayname string  //Display name
name string  //Processor name
locked int  //Is Locked by admin?
lockedmessage string  Optional //Text to display if locked
userconfigured int  //Is configured?
loggedin  //DEPRECATED ATTRIBUTE -
                                         Kept for backward compatibility, use enabled instead.
object {
name string  //Name
displayname string  //Display name
checked int  //Is checked?
})
loggedoff  //DEPRECATED ATTRIBUTE -
                                         Kept for backward compatibility, use enabled instead.
object {
name string  //Name
displayname string  //Display name
checked int  //Is checked?
)
enabled int  //Is enabled?
)
))
))
blocknoncontacts int  //Privacy messaging setting to define who can message you
entertosend int  //User preference for using enter to send messages
warnings  Optional //list of warnings
list of (
 //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
)
))
}

```



XML-RPC (PHP structure)

```

Array
(
    [preferences] =>
        Array
            (
                [userid] => int
                [disableall] => int
                [processors] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [displayname] => string
                                        [name] => string
                                        [hassettings] => int
                                        [contextid] => int
                                        [userconfigured] => int
                                    )
                        )
                )
            )
    [components] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [displayname] => string
                            [notifications] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [displayname] => string
                                                    [preferencekey] => string
                                                    [processors] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                            [displayname] => string
                                                                            [name] => string
                                                                            [locked] => int
                                                                            [lockedmessage] => string
                                                                            [userconfigured] => int
                                                                            [loggedin] =>
                                                                                Array
                                                                                    (
                                                                                        [name] => string
                                                                                        [displayname] => string
                                                                                        [checked] => int
                                                                                    )
                                                                                ^
                                                                            [loggedoff] =>
                                                                                Array
                                                                                    (
                                                                                        [name] => string
                                                                                        [displayname] => string
                                                                                        [checked] => int
                                                                                    )
                                                                                ^
                                                                            [enabled] => int
                                                                        )
                                                                )
                                                        )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
)
[blocknoncontacts] => int
[entertosend] => int
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
        )
)

```



^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="preferences">
      <SINGLE>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="disableall">
          <VALUE>int</VALUE>
        </KEY>
      <KEY name="processors">
        <MULTIPLE>
          <SINGLE>
            <KEY name="displayname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="hassettings">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="contextid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userconfigured">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="components">
        <MULTIPLE>
          <SINGLE>
            <KEY name="displayname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="notifications">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="displayname">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="preferencekey">
                    <VALUE>string</VALUE>
                  </KEY>
                </SINGLE>
              </MULTIPLE>
            <KEY name="processors">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="displayname">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="name">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="locked">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="lockedmessage">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="userconfigured">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="loggedin">
                    <SINGLE>
                      <KEY name="name">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="displayname">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="checked">
                        <VALUE>int</VALUE>
                      </KEY>
                    </SINGLE>
                  </KEY>
                  <KEY name="loggedoff">
                    <SINGLE>
                      <KEY name="name">
```



```

<VALUE>string</VALUE>
</KEY>
<KEY name="displayname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="checked">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="enabled">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="blocknoncontacts">
    <VALUE>int</VALUE>
</KEY>
<KEY name="entertosend">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_get_user_notification_preferences

Get the notification preferences for a given user.

Arguments

userid (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response



General structure

```

object {
preferences object {
userid int    //User id
disableall int   //Whether all the preferences are disabled
processors  //Config form values
list of (
object {
displayname string  //Display name
name string  //Processor name
hassettings int   //Whether has settings
contextid int    //Context id
userconfigured int //Whether is configured by the user
})
components  //Available components
list of (
object {
displayname string  //Display name
notifications //List of notifications for the component
list of (
object {
displayname string  //Display name
preferencekey string  //Preference key
processors  //Processors values for this notification
list of (
object {
displayname string  //Display name
name string  //Processor name
locked int    //Is Locked by admin?
lockedmessage string  Optional //Text to display if locked
userconfigured int //Is configured?
loggedin  //DEPRECATED ATTRIBUTE -
                                         Kept for backward compatibility, use enabled instead.
object {
name string  //Name
displayname string  //Display name
checked int   //Is checked?
})
loggedoff  //DEPRECATED ATTRIBUTE -
                                         Kept for backward compatibility, use enabled instead.
object {
name string  //Name
displayname string  //Display name
checked int   //Is checked?
)
enabled int   //Is enabled?
)}
)})
})
})
warnings Optional //List of warnings
list of (
 //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
})
)
}

```



XML-RPC (PHP structure)

```

Array
(
    [preferences] =>
        Array
            (
                [userid] => int
                [disableall] => int
                [processors] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [displayname] => string
                                        [name] => string
                                        [hassettings] => int
                                        [contextid] => int
                                        [userconfigured] => int
                                    )
                        )
                )
            )
    [components] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [displayname] => string
                            [notifications] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [displayname] => string
                                                    [preferencekey] => string
                                                    [processors] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                            [displayname] => string
                                                                            [name] => string
                                                                            [locked] => int
                                                                            [lockedmessage] => string
                                                                            [userconfigured] => int
                                                                            [loggedin] =>
                                                                                Array
                                                                                    (
                                                                                        [name] => string
                                                                                        [displayname] => string
                                                                                        [checked] => int
                                                                                    )
                                                                                ^
                                                                            [loggedoff] =>
                                                                                Array
                                                                                    (
                                                                                        [name] => string
                                                                                        [displayname] => string
                                                                                        [checked] => int
                                                                                    )
                                                                                ^
                                                                            [enabled] => int
                                                                        )
                                                                )
                                                        )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
)
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
            )
        )
)

```



^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="preferences">
      <SINGLE>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="disableall">
          <VALUE>int</VALUE>
        </KEY>
      <KEY name="processors">
        <MULTIPLE>
          <SINGLE>
            <KEY name="displayname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="hassettings">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="contextid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="userconfigured">
              <VALUE>int</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="components">
        <MULTIPLE>
          <SINGLE>
            <KEY name="displayname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="notifications">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="displayname">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="preferencekey">
                    <VALUE>string</VALUE>
                  </KEY>
                </SINGLE>
              </MULTIPLE>
            <KEY name="processors">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="displayname">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="name">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="locked">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="lockedmessage">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="userconfigured">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="loggedin">
                    <SINGLE>
                      <KEY name="name">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="displayname">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="checked">
                        <VALUE>int</VALUE>
                      </KEY>
                    </SINGLE>
                  </KEY>
                  <KEY name="loggedoff">
                    <SINGLE>
                      <KEY name="name">
```



```

        <VALUE>string</VALUE>
    </KEY>
    <KEY name="displayname">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="checked">
        <VALUE>int</VALUE>
    </KEY>
    </SINGLE>
    </KEY>
    <KEY name="enabled">
        <VALUE>int</VALUE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_message_mark_all_conversation_messages_as_read

Mark all conversation messages as read for a given user

Arguments

userid (Required)

The user id who we are marking the messages as read for

General structure

```
int //The user id who we are marking the messages as read for
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversationid (Required)

The conversation id who we are marking the messages as read for

General structure

```
int //The conversation id who we are marking the messages as read for
```

XML-RPC (PHP structure)

```
[conversationid] => int
```

REST (POST parameters)

```
conversationid= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_mark_all_notifications_as_read

Mark all notifications as read for a given user

Arguments

useridto (Required)

the user id who received the message, 0 for any user

General structure

```
int //the user id who received the message, 0 for any user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```

useridfrom (Default to "0")

the user id who send the message, 0 for any user. -10 or -20 for no-reply or support user

General structure

```
int Default to "0" //the user id who send the message, 0 for any user. -10 or -20 for no-reply or support user
```

XML-RPC (PHP structure)

```
[useridfrom] => int
```

REST (POST parameters)

```
useridfrom= int
```

timecreatedto (Default to "0")

mark messages created before this time as read, 0 for all messages

General structure

```
int Default to "0" //mark messages created before this time as read, 0 for all messages
```

XML-RPC (PHP structure)

```
[timecreatedto] => int
```

REST (POST parameters)

```
timecreatedto= int
```

**Response**

True if the messages were marked read, false otherwise

General structure

```
int //True if the messages were marked read, false otherwise
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_mark_message_read

Mark a single message as read, trigger message_viewed event.

Arguments**messageid** (Required)

id of the message in the messages table

General structure

```
int //id of the message in the messages table
```

XML-RPC (PHP structure)

```
[messageid] => int
```

**REST (POST parameters)**

```
messageid= int
```

timeread (Default to "0")

timestamp for when the message should be marked read

General structure

```
int Default to "0" //timestamp for when the message should be marked read
```

XML-RPC (PHP structure)

```
[timeread] => int
```

REST (POST parameters)

```
timeread= int
```

Response**General structure**

```
object {
    messageid int //the id of the message in the messages table
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [messageid] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="messageid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_mark_notification_read

Mark a single notification as read, trigger notification_viewed event.

Arguments

notificationid (Required)

id of the notification

General structure

```
int //id of the notification
```

XML-RPC (PHP structure)

```
[notificationid] => int
```

REST (POST parameters)

```
notificationid= int
```

timeread (Default to "0")

timestamp for when the notification should be marked read

General structure

```
int Default to "0" //timestamp for when the notification should be marked read
```

XML-RPC (PHP structure)

```
[timeread] => int
```

REST (POST parameters)

```
timeread= int
```

Response

General structure

```

object {
notificationid int //id of the notification
warnings Optional //list of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}

```

XML-RPC (PHP structure)

```

Array
(
[notificationid] => int
[warnings] =>
    Array
    (
[0] =>
        Array
        (
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="notificationid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_message_processor_config_form

Process the message processor config form

Arguments

userid (Required)

id of the user, 0 for current user

General structure

```
int //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

name (Required)

The name of the message processor

General structure

```
string //The name of the message processor
```

XML-RPC (PHP structure)

```
[name] => string
```

REST (POST parameters)

```
name= string
```

formvalues (Required)

Config form values

General structure

```
//Config form values
list of (
object {
name string //name of the form element
value string //value of the form element
}
)
```

XML-RPC (PHP structure)

```
[formvalues] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [name] => string
                        [value] => string
                    )
                )
        )
    )
```

REST (POST parameters)

```
formvalues[0][name]= string
formvalues[0][value]= string
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_message_search_users

Retrieve the data for searching for people

**Arguments****userid** (Required)

The id of the user who is performing the search

General structure

```
int //The id of the user who is performing the search
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

search (Required)

The string being searched

General structure

```
string //The string being searched
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

limitfrom (Default to "0")

Limit from

General structure

```
int Default to "0" //Limit from
```

XML-RPC (PHP structure)

```
[limitfrom] => int
```

REST (POST parameters)

```
limitfrom= int
```

limitnum (Default to "0")

Limit number

General structure

```
int Default to "0" //Limit number
```

XML-RPC (PHP structure)

```
[limitnum] => int
```

REST (POST parameters)

```
limitnum= int
```

**Response**

General structure

```

object {
contacts list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
shownlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)
)noncontacts list of (
object {
id int //The user id
fullname string //The user's name
profileurl string //The link to the user's profile page
profileimageurl string //User picture URL
profileimageurlsmall string //Small user picture URL
isonline int //The user's online status
shownlinestatus int //Show the user's online status?
isblocked int //If the user has been blocked
iscontact int //Is the user a contact?
isdeleted int //Is the user deleted?
canmessageevenifblocked int //If the user can still message even if they get blocked
canmessage int //If the user can be messaged
requirescontact int //If the user requires to be contacts
contactrequests Optional //The contact requests
list of (
object {
id int //The id of the contact request
userid int //The id of the user who created the contact request
requesteduserid int //The id of the user confirming the request
timecreated int //The timecreated timestamp for the contact request
})
)conversations Optional //Conversations between users
list of (
Optional //information about conversation
object {
id int //Conversations id
type int //Conversation type: private or public
name string //Multilang compatible conversation name2
timecreated int //The timecreated timestamp for the conversation
})
)
)
)
}

```



XML-RPC (PHP structure)

```

Array
(
    [contacts] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [fullname] => string
                            [profileurl] => string
                            [profileimageurl] => string
                            [profileimageurlsmall] => string
                            [isonline] => int
                            [showonlinestatus] => int
                            [isblocked] => int
                            [iscontact] => int
                            [isdeleted] => int
                            [canmessageevenifblocked] => int
                            [canmessage] => int
                            [requirescontact] => int
                            [contactrequests] =>
                                Array
                                (
                                    [
                                        [0] =>
                                            Array
                                            (
                                                [
                                                    [id] => int
                                                    [userid] => int
                                                    [requesteduserid] => int
                                                    [timecreated] => int
                                                )
                                            )
                                    )
                                )
                            [conversations] =>
                                Array
                                (
                                    [
                                        [0] =>
                                            Array
                                            (
                                                [
                                                    [id] => int
                                                    [type] => int
                                                    [name] => string
                                                    [timecreated] => int
                                                )
                                            )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
[noncontacts] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [fullname] => string
                        [profileurl] => string
                        [profileimageurl] => string
                        [profileimageurlsmall] => string
                        [isonline] => int
                        [showonlinestatus] => int
                        [isblocked] => int
                        [iscontact] => int
                        [isdeleted] => int
                        [canmessageevenifblocked] => int
                        [canmessage] => int
                        [requirescontact] => int
                        [contactrequests] =>
                            Array
                            (
                                [
                                    [0] =>
                                        Array
                                        (
                                            [
                                                [
                                                    [id] => int
                                                    [userid] => int
                                                    [requesteduserid] => int
                                                    [timecreated] => int
                                                )
                                            )
                                        )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
[conversations] =>
    Array

```

```
{  
    [0] =>  
        Array  
            (  
                [id] => int  
                [type] => int  
                [name] => string  
                [timecreated] => int  
            )  
    )  
}  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="contacts">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurlsmall">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isonline">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="showonlinestatus">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isblocked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="iscontact">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isdeleted">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="canmessageevenifblocked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="canmessage">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="requirescontact">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="contactrequests">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="requesteduserid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                    <KEY name="conversations">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="type">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

```
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="noncontacts">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="fullname">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
<VALUE>string</VALUE>
</KEY>
<KEY name="isononline">
<VALUE>int</VALUE>
</KEY>
<KEY name="showonlinestatus">
<VALUE>int</VALUE>
</KEY>
<KEY name="isblocked">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscontact">
<VALUE>int</VALUE>
</KEY>
<KEY name="isdeleted">
<VALUE>int</VALUE>
</KEY>
<KEY name="canmessageevenifblocked">
<VALUE>int</VALUE>
</KEY>
<KEY name="canmessage">
<VALUE>int</VALUE>
</KEY>
<KEY name="requirescontact">
<VALUE>int</VALUE>
</KEY>
<KEY name="contactrequests">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="requesteduserid">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="conversations">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="type">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
```



```
</KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_mute_conversations

Mutes a list of conversations

Arguments**userid** (Required)

The id of the user who is blocking

General structure

```
int //The id of the user who is blocking
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversationids (Required)

General structure

```
list of (
int //id of the conversation
)
```

XML-RPC (PHP structure)

```
[conversationids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
conversationids[0]= int
```

Response

list of warnings

General structure

```
Optional //list of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
        Array
        (
            [
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            ]
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`core_message_search_contacts`

Search for contacts

Arguments

`searchtext` (Required)

String the user's fullname has to match to be found

General structure

```
string //String the user's fullname has to match to be found
```

XML-RPC (PHP structure)

```
[searchtext] => string
```

REST (POST parameters)

```
searchtext= string
```

`onlymycourses` (Default to "")

Limit search to the user's courses

General structure

```
int Default to "" //Limit search to the user's courses
```

XML-RPC (PHP structure)

```
[onlymycourses] => int
```

REST (POST parameters)

```
onlymycourses= int
```



Response

List of contacts

General structure

```
//List of contacts
list of (
object {
id int //User ID
fullname string //User full name
profileimageurl string Optional //User picture URL
profileimageurlsmall string Optional //Small user picture URL
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [id] => int
                    [fullname] => string
                    [profileimageurl] => string
                    [profileimageurlsmall] => string
                )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimageurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimageurlsmall">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

core_message_send_instant_messages

Send instant messages

Arguments**messages** (Required)

General structure

```
list of (
object {
touserid int //id of the user to send the private message
text string //the text of the message
textformat int Default to "0" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
clientmsgid string Optional //your own client id for the message. If this id is provided, the fail message id will be returned to you
}
)
```

XML-RPC (PHP structure)

```
[messages] =>
Array
(
[0] =>
Array
(
[touserid] => int
[text] => string
[textformat] => int
[clientmsgid] => string
)
)
```

REST (POST parameters)

```
messages[0][touserid]= int
messages[0][text]= string
messages[0][textformat]= int
messages[0][clientmsgid]= string
```

Response**General structure**

```
list of (
object {
msgid int //test this to know if it succeeds: id of the created message if it succeeded, -1 when failed
clientmsgid string Optional //your own id for the message
errormessage string Optional //error message - if it failed
text string Optional //The text of the message
timecreated int Optional //The timecreated timestamp for the message
conversationid int Optional //The conversation id for this message
useridfrom int Optional //The user id who sent the message
candeletemessagesforallusers int Default to "" //If the user can delete messages in the conversation for all users
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
Array
(
[msgid] => int
[clientmsgid] => string
[errormessage] => string
[text] => string
[timecreated] => int
[conversationid] => int
[useridfrom] => int
[candeletemessagesforallusers] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="msgid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="clientmsgid">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="errormessage">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="text">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="conversationid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="useridfrom">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="candeletemessagesforallusers">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

Yes

core_message_send_messages_to_conversation

Send messages to an existing conversation between users

Arguments**conversationid** (Required)

id of the conversation

General structure

```
int //id of the conversation
```

XML-RPC (PHP structure)

```
[conversationid] => int
```

REST (POST parameters)

```
conversationid= int
```

messages (Required)

General structure

```
list of (
object {
text string //the text of the message
textformat int Default to "0" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
```

XML-RPC (PHP structure)

```
[messages] =>
Array
(
[0] =>
Array
(
[text] => string
[textformat] => int
)
)
```

REST (POST parameters)

```
messages[0][text]= string
messages[0][textformat]= int
```

Response**General structure**

```
list of (
object {
id int //The id of the message
useridfrom int //The id of the user who sent the message
text string //The text of the message
timecreated int //The timecreated timestamp for the message
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
Array
(
[id] => int
[useridfrom] => int
[text] => string
[timecreated] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="useridfrom">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_set_favourite_conversations

Mark a conversation or group of conversations as favourites/starred conversations.

**Arguments****userid** (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversations (Required)

General structure

```
list of (
int Default to "0" //id of the conversation
)
```

XML-RPC (PHP structure)

```
[conversations] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
conversations[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="item">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
<VALUE>string</VALUE>
</KEY>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_unblock_user

Unblocks a user

Arguments

userid (Required)

The id of the user who is unblocking

General structure

```
int //The id of the user who is unblocking
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

unblockeduserid (Required)

The id of the user being unblocked

General structure

```
int //The id of the user being unblocked
```

XML-RPC (PHP structure)

```
[unblockeduserid] => int
```

REST (POST parameters)

```
unblockeduserid= int
```

Response

list of warnings

General structure

```

Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
        (
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_message_unmute_conversations

Unmutes a list of conversations

Arguments**userid** (Required)

The id of the user who is unblocking

General structure

```
int //The id of the user who is unblocking
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversationids (Required)

General structure

```
list of (
int //id of the conversation
)
```

XML-RPC (PHP structure)

```
[conversationids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
conversationids[0]= int
```

**Response**

list of warnings

General structure

```
Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_message_unset_favourite_conversations

Unset a conversation or group of conversations as favourites/starred conversations.

Arguments**userid** (Default to "0")

id of the user, 0 for current user

General structure

```
int Default to "0" //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

conversations (Required)

General structure

```
list of (
int Default to "0" //id of the conversation
)
```

XML-RPC (PHP structure)

```
[conversations] =>
    Array
        (
            [0] => int
        )
)
```

REST (POST parameters)

```
conversations[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
    [0] =>
        Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_moodlenet_auth_check

Check a user has authorized for a given MoodleNet site

**Arguments**

issuerid (Required)

OAuth 2 issuer ID

General structure

```
int //OAuth 2 issuer ID
```

XML-RPC (PHP structure)

```
[issuerid] => int
```

REST (POST parameters)

```
issuerid= int
```

courseid (Required)

Course ID

General structure

```
int //Course ID
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
    loginurl string //Login url
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
    }
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
}
```

XML-RPC (PHP structure)

```
Array
(
    [loginurl] => string
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="loginurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_moodlenet_get_share_info_activity

Get information about an activity being shared

Arguments**cmid** (Required)

The cmid of the activity

General structure

```
int //The cmid of the activity
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

Response

General structure

```
object {
    name string //Activity name
    type string //Activity type
    server string //MoodleNet server
    supportpageurl string //Support page URL
    issuerid int //MoodleNet issuer id
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [name] => string
    [type] => string
    [server] => string
    [supportpageurl] => string
    [issuerid] => int
    [status] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="type">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="server">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="supportpageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="issuerid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_moodlenet_send_activity

Send activity to MoodleNet

Arguments

issuerid (Required)

OAuth 2 issuer ID

General structure

```
int //OAuth 2 issuer ID
```

XML-RPC (PHP structure)

```
[issuerid] => int
```

REST (POST parameters)

```
issuerid= int
```

cmid (Required)

Course module ID

General structure

```
int //Course module ID
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

shareformat (Required)

Share format

General structure

```
int //Share format
```

XML-RPC (PHP structure)

```
[shareformat] => int
```

REST (POST parameters)

```
shareformat= int
```

**Response****General structure**

```
object {
  status int //Status: true if success
  resourceurl string //Resource URL from MoodleNet
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [resourceurl] => string
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="resourceurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_notes_create_notes**Create notes****Arguments**

notes (Required)

General structure

```
list of (
object {
userid int //id of the user the note is about
publishstate string //personal', 'course' or 'site'
courseid int //course id of the note (in Moodle a note can only be created into a course, even for site and personal notes)
text string //the text of the message - text or HTML
format int Default to "1" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
clientnoteid string Optional //your own client id for the note. If this id is provided, the fail message id will be returned to you
}
)
```

XML-RPC (PHP structure)

```
[notes] =>
Array
(
[0] =>
Array
(
[userid] => int
[publishstate] => string
[courseid] => int
[text] => string
[format] => int
[clientnoteid] => string
)
)
```

REST (POST parameters)

```
notes[0][userid]= int
notes[0][publishstate]= string
notes[0][courseid]= int
notes[0][text]= string
notes[0][format]= int
notes[0][clientnoteid]= string
```

**Response****General structure**

```
list of (
object {
clientnoteid string Optional //your own id for the note
noteid int //ID of the created note when successful, -1 when failed
errormessage string Optional //error message - if failed
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [clientnoteid] => string
                    [noteid] => int
                    [errormessage] => string
                ]
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="clientnoteid">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="noteid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="errormessage">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

**Callable from AJAX**

Yes

core_notes_delete_notes**Delete notes****Arguments****notes** (Required)

Array of Note Ids to be deleted.

General structure

```
//Array of Note Ids to be deleted.
list of (
    int //ID of the note to be deleted
)
```

XML-RPC (PHP structure)

```
[notes] =>
    Array
    (
        [
            [0] => int
        )
    )
```

REST (POST parameters)

```
notes[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
object {
item string Optional //item is always 'note'
itemid int Optional //When errorcode is savedfailed the note could not be modified. When errorcode is badparam, an incorrect parameter was provided. When errorcode is badid, the note does not exist
warningcode string //errorcode can be badparam (incorrect parameter), savedfailed (could not be modified), or badid (note does not exist)
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_notes_get_course_notes

Returns all notes in specified course (or site), for the specified user.

Arguments**courseid** (Required)

course id, 0 for SITE

General structure

```
int //course id, 0 for SITE
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

user id

General structure

```
int Default to "0" //user id
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

notes

```

General structure

//notes
object {
sitenotes Optional //site notes
list of (
object {
id int //id of this note
courseid int //id of the course
userid int //user id
content string //the content text formated
format int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
created int //time created (timestamp)
lastmodified int //time of last modification (timestamp)
usermodified int //user id of the creator of this note
publishstate string //state of the note (i.e. draft, public, site)
}
)coursenotes Optional //couse notes
list of (
object {
id int //id of this note
courseid int //id of the course
userid int //user id
content string //the content text formated
format int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
created int //time created (timestamp)
lastmodified int //time of last modification (timestamp)
usermodified int //user id of the creator of this note
publishstate string //state of the note (i.e. draft, public, site)
}
)personalnotes Optional //personal notes
list of (
object {
id int //id of this note
courseid int //id of the course
userid int //user id
content string //the content text formated
format int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
created int //time created (timestamp)
lastmodified int //time of last modification (timestamp)
usermodified int //user id of the creator of this note
publishstate string //state of the note (i.e. draft, public, site)
}
)canmanagesystemnotes int Optional //Whether the user can manage notes at system level.
canmanagecoursenotes int Optional //Whether the user can manage notes at the given course.
warnings Optional //list of warnings
list of (
 /warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)
}

```



XML-RPC (PHP structure)

```
Array
(
    [sitenotes] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [courseid] => int
                            [userid] => int
                            [content] => string
                            [format] => int
                            [created] => int
                            [lastmodified] => int
                            [usermodified] => int
                            [publishstate] => string
                        )
                )
            )
    [coursenotes] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [courseid] => int
                            [userid] => int
                            [content] => string
                            [format] => int
                            [created] => int
                            [lastmodified] => int
                            [usermodified] => int
                            [publishstate] => string
                        )
                )
            )
    [personalnotes] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [courseid] => int
                            [userid] => int
                            [content] => string
                            [format] => int
                            [created] => int
                            [lastmodified] => int
                            [usermodified] => int
                            [publishstate] => string
                        )
                )
            )
    [canmanagesystemnotes] => int
    [canmanagecoursenotes] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="sitenotes">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="content">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="created">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lastmodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usermodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="publishstate">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    <KEY name="coursenotes">
        <MULTIPLE>
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="courseid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="content">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="format">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="created">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="lastmodified">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="usermodified">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="publishstate">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
<KEY name="personalnotes">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

<KEY name="content">
    <VALUE>string</VALUE>
</KEY>
<KEY name="format">
    <VALUE>int</VALUE>
</KEY>
<KEY name="created">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lastmodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="publishstate">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="canmanagesystemnotes">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canmanagecoursenotes">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_notes_get_notes

Get notes

Arguments**notes** (Required)

Array of Note Ids to be retrieved.

General structure

```
//Array of Note Ids to be retrieved.
list of (
int //ID of the note to be retrieved
)
```

XML-RPC (PHP structure)

```
[notes] =>
  Array
    (
      [0] => int
    )
```

REST (POST parameters)

```
notes[0]= int
```

Response**General structure**

```
object {
notes list of (
 //note
object {
noteid int Optional //id of the note
userid int Optional //id of the user the note is about
publishstate string Optional //'personal', 'course' or 'site'
courseid int Optional //course id of the note
text string Optional //the text of the message - text or HTML
format int Optional //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item is always 'note'
itemid int Optional //When errorcode is savedfailed the note could not be modified. When errorcode is badparam, an incorrect parameter was provided. When errorcode is badid, the note does not exist
warningcode string //errorcode can be badparam (incorrect parameter), savedfailed (could not be modified), or badid (note does not exist)
message string //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```
Array
(
    [notes] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [noteid] => int
                            [userid] => int
                            [publishstate] => string
                            [courseid] => int
                            [text] => string
                            [format] => int
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="notes">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="noteid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="publishstate">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="text">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="format">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_notes_update_notes**Update notes**

Arguments

`notes` (Default to "Array ()")

Array of Notes

```
General structure

Default to "Array
(
)
" //Array of Notes
list of (
object {
id int //id of the note
publishstate string //'personal', 'course' or 'site'
text string //the text of the message - text or HTML
format int Default to "1" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
```

XML-RPC (PHP structure)

```
[notes] =>
Array
(
[0] =>
Array
(
[id] => int
[publishstate] => string
[text] => string
[format] => int
)
)
```

REST (POST parameters)

```
notes[0][id]= int
notes[0][publishstate]= string
notes[0][text]= string
notes[0][format]= int
```

Response

list of warnings

```
General structure

Optional //List of warnings
list of (
//warning
object {
item string Optional //item is always 'note'
itemid int Optional //When errorcode is savedfailed the note could not be modified. When errorcode is badparam, an incorrect parameter was provided. When errorcode is badid, the note does not exist
warningcode string //errorcode can be badparam (incorrect parameter), savedfailed (could not be modified), or badid (note does not exist)
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

core_notes_view_notes

Simulates the web interface view of notes/index.php: trigger events.

Arguments**courseid** (Required)

course id, 0 for notes at system level

General structure

```
int //course id, 0 for notes at system level
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

user id, 0 means view all the user notes

General structure

```
int Default to "0" //user id, 0 means view all the user notes
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
}
```

**XML-RPC (PHP structure)**

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

[core_output_load_fontawesome_icon_map](#) (Deprecated)

Load the mapping of names to icons

Arguments**Response****General structure**

```
list of (
object {
component string //The component for the icon.
pix string //Value to map the icon from.
to string //Value to map the icon to.
})
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [component] => string
                    [pix] => string
                    [to] => string
                ]
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="pix">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="to">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

**Callable from AJAX**

Yes

core_output_load_fontawesome_icon_system_map

Load the mapping of moodle pix names to fontawesome icon names

Arguments**themename** (Required)

The theme to fetch the map for

General structure

```
string //The theme to fetch the map for
```

XML-RPC (PHP structure)

```
[themename] => string
```

REST (POST parameters)

```
themename= string
```

Response**General structure**

```
list of (
object {
component string //The component for the icon.
pix string //Value to map the icon from.
to string //Value to map the icon to.
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[component] => string
[pix] => string
[to] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="component">
<VALUE>string</VALUE>
</KEY>
<KEY name="pix">
<VALUE>string</VALUE>
</KEY>
<KEY name="to">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

core_output_load_template

Load a template for a renderable

Arguments**component** (Required)

component containing the template

General structure`string //component containing the template`**XML-RPC (PHP structure)**`[component] => string`**REST (POST parameters)**`component= string`**template** (Required)

name of the template

General structure`string //name of the template`**XML-RPC (PHP structure)**`[template] => string`**REST (POST parameters)**`template= string`**themename** (Required)

The current theme.

General structure`string //The current theme.`**XML-RPC (PHP structure)**`[themename] => string`**REST (POST parameters)**`themename= string`**includecomments** (Default to "")

Include comments or not

General structure

```
int Default to "" //Include comments or not
```

XML-RPC (PHP structure)

```
[includecomments] => int
```

REST (POST parameters)

```
includecomments= int
```

Response

template

General structure

```
string //template
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

Callable from AJAX

Yes

core_output_load_template_with_dependencies

Load a template and its dependencies for a renderable

Arguments

component (Required)

component containing the template

General structure

```
string //component containing the template
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

template (Required)

name of the template

General structure

```
string //name of the template
```

XML-RPC (PHP structure)

```
[template] => string
```

REST (POST parameters)

```
template= string
```

themename (Required)

The current theme.

General structure

```
string //The current theme.
```

XML-RPC (PHP structure)

```
[themename] => string
```

REST (POST parameters)

```
themename= string
```

**includecomments** (Default to "")

Include comments or not

General structure

```
int Default to "" //Include comments or not
```

XML-RPC (PHP structure)

```
[includecomments] => int
```

REST (POST parameters)

```
includecomments= int
```

lang (Default to "null")

lang

General structure

```
string Default to "null" //Lang
```

XML-RPC (PHP structure)

```
[lang] => string
```

REST (POST parameters)

```
lang= string
```

Response**General structure**

```
object {
templates list of (
object {
component string //component containing the resource
name string //name of the resource
value string //resource value
})
)strings list of (
object {
component string //component containing the resource
name string //name of the resource
value string //resource value
})
)}
```

XML-RPC (PHP structure)

```
Array
(
    [
        [templates] =>
            Array
                (
                    [
                        [0] =>
                            Array
                                (
                                    [
                                        [component] => string
                                        [name] => string
                                        [value] => string
                                    )
                                )
                )
        [strings] =>
            Array
                (
                    [
                        [0] =>
                            Array
                                (
                                    [
                                        [component] => string
                                        [name] => string
                                        [value] => string
                                    )
                                )
                )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="templates">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="strings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

Callable from AJAX

Yes

core_payment_get_available_gateways

Get the list of payment gateways that support the given component/area

Arguments

component (Required)

Component

General structure

```
string //Component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

paymentarea (Required)

Payment area in the component

General structure

```
string //Payment area in the component
```

XML-RPC (PHP structure)

```
[paymentarea] => string
```

REST (POST parameters)

```
paymentarea= string
```

itemid (Required)

An identifier for payment area in the component

General structure

```
int //An identifier for payment area in the component
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

**Response****General structure**

```
list of (
object {
shortname string //Name of the plugin
name string //Human readable name of the gateway
description string //description of the gateway
surcharge int //percentage of surcharge when using the gateway
cost string //Cost in human-readable form (amount plus surcharge with currency sign)
})
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [shortname] => string
                [name] => string
                [description] => string
                [surcharge] => int
                [cost] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="surcharge">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="cost">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_question_get_random_question_summaries

Get the random question set for a criteria

Arguments**categoryid** (Required)

Category id to find random questions

General structure

```
int //Category id to find random questions
```

XML-RPC (PHP structure)

```
[categoryid] => int
```

REST (POST parameters)

```
categoryid= int
```

includesubcategories (Required)

Include the subcategories in the search

General structure

```
int //Include the subcategories in the search
```

XML-RPC (PHP structure)

```
[includesubcategories] => int
```

REST (POST parameters)

```
includesubcategories= int
```

tagids (Required)**General structure**

```
list of (
int //Tag id
)
```

XML-RPC (PHP structure)

```
[tagids] =>
  Array
    (
      [0] => int
    )
```

**REST (POST parameters)**

```
tagids[0]= int
```

contextid (Required)

Context id that the questions will be rendered in (used for exporting)

General structure

```
int //Context id that the questions will be rendered in (used for exporting)
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

limit (Default to "0")

Maximum number of results to return

General structure

```
int Default to "0" //Maximum number of results to return
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

offset (Default to "0")

Number of items to skip from the beginning of the result set

General structure

```
int Default to "0" //Number of items to skip from the beginning of the result set
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

Response**General structure**

```
object {
    totalcount int //total number of questions in result set
    questions list of (
        object {
            id int //id
            category int //category
            parent int //parent
            name string //name
            qtype string //qtype
            icon object {
                key string //key
                component string //component
                alttext string //alttext
            }
        }
    )
}
```



XML-RPC (PHP structure)

```
Array
(
    [totalcount] => int
    [questions] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [category] => int
                            [parent] => int
                            [name] => string
                            [qtype] => string
                            [icon] =>
                                Array
                                (
                                    [
                                        [key] => string
                                        [component] => string
                                        [alttext] => string
                                    )
                                )
                            )
                        )
                    )
            )
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="totalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="questions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="category">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="parent">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="qtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="icon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="alttext">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

5

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_question_submit_tags_form ([Deprecated](#))

Update the question tags.

Arguments**questionid** (Required)

The question id

General structure

```
int //The question id
```

XML-RPC (PHP structure)

```
[questionid] => int
```

REST (POST parameters)

```
questionid= int
```

contextid (Required)

The editing context id

General structure

```
int //The editing context id
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

formdata (Required)

The data from the tag form

General structure

```
string //The data from the tag form
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response**General structure**

```
object {
    status int //status: true if success
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_question_update_flag

Update the flag state of a question attempt.

Arguments

qubaid (Required)

the question usage id.

General structure

```
int //the question usage id.
```

XML-RPC (PHP structure)

```
[qubaid] => int
```

REST (POST parameters)

```
qubaid= int
```

questionid (Required)

the question id

General structure

```
int //the question id
```

XML-RPC (PHP structure)

```
[questionid] => int
```

REST (POST parameters)

```
questionid= int
```

qaid (Required)

the question_attempt id

General structure

```
int //the question_attempt id
```

XML-RPC (PHP structure)

```
[qaid] => int
```

REST (POST parameters)

```
qaid= int
```



slot (Required)

the slot number within the usage

General structure

```
int //the slot number within the usage
```

XML-RPC (PHP structure)

```
[slot] => int
```

REST (POST parameters)

```
slot= int
```

checksum (Required)

computed checksum with the last three arguments and the users username

General structure

```
string //computed checksum with the last three arguments and
      the users username
```

XML-RPC (PHP structure)

```
[checksum] => string
```

REST (POST parameters)

```
checksum= string
```

newstate (Required)

the new state of the flag. true = flagged

General structure

```
int //the new state of the flag. true = flagged
```

XML-RPC (PHP structure)

```
[newstate] => int
```

REST (POST parameters)

```
newstate= int
```

Response**General structure**

```
object {
  status int //status: true if success
  warnings Optional //list of warnings
  list of (
    /warning
  )
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
}
```

**XML-RPC (PHP structure)**

```
Array
(
  [status] => int
  [warnings] =>
    Array
    (
      [0] =>
        Array
        (
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**core_rating_add_rating**

Rates an item.

Arguments**contextlevel** (Required)

context level: course, module, user, etc...

General structure

```
string //context Level: course, module, user, etc...
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Required)

the instance id of item associated with the context level

General structure

```
int //the instance id of item associated with the context Level
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

ratingarea (Required)

rating area

General structure

```
string //rating area
```

**XML-RPC (PHP structure)**

```
[ratingarea] => string
```

REST (POST parameters)

```
ratingarea= string
```

itemid (Required)

associated id

General structure

```
int //associated id
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

scaleid (Required)

scale id

General structure

```
int //scale id
```

XML-RPC (PHP structure)

```
[scaleid] => int
```

REST (POST parameters)

```
scaleid= int
```

rating (Required)

user rating

General structure

```
int //user rating
```

XML-RPC (PHP structure)

```
[rating] => int
```

REST (POST parameters)

```
rating= int
```

rateduserid (Required)

rated user id

General structure

```
int //rated user id
```

**XML-RPC (PHP structure)**

```
[rateduserid] => int
```

REST (POST parameters)

```
rateduserid= int
```

aggregation (Default to "0")

aggregation method

General structure

```
int Default to "0" //aggregation method
```

XML-RPC (PHP structure)

```
[aggregation] => int
```

REST (POST parameters)

aggregation= int

Response**General structure**

```
object {
    success int //Whether the rate was successfully created
    aggregate string Optional //New aggregate
    count int Optional //Ratings count
    itemid int Optional //Rating item id
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [success] => int
    [aggregate] => string
    [count] => int
    [itemid] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="success">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="aggregate">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

core_rating_get_item_ratings

Retrieve all the ratings for an item.

Arguments

contextlevel (Required)

context level: course, module, user, etc...

General structure

```
string //context Level: course, module, user, etc...
```

XML-RPC (PHP structure)

```
[contextlevel] => string
```

REST (POST parameters)

```
contextlevel= string
```

instanceid (Required)

the instance id of item associated with the context level

General structure

```
int //the instance id of item associated with the context Level
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

component (Required)

component

General structure

```
string //component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

**ratingarea (Required)**

rating area

General structure

```
string //rating area
```

XML-RPC (PHP structure)

```
[ratingarea] => string
```

REST (POST parameters)

```
ratingarea= string
```

itemid (Required)

associated id

General structure

```
int //associated id
```

XML-RPC (PHP structure)

```
[itemid] => int
```

REST (POST parameters)

```
itemid= int
```

scaleid (Required)

scale id

General structure

```
int //scale id
```

XML-RPC (PHP structure)

```
[scaleid] => int
```

REST (POST parameters)

```
scaleid= int
```

sort (Required)

sort order (firstname, rating or timemodified)

General structure

```
string //sort order (firstname, rating or timemodified)
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```



Response

General structure

```

object {
  ratings //list of ratings
  list of (
    //Rating
    object {
      id int //rating id
      userid int //user id
      userpictureurl string //URL user picture
      userfullname string //user fullname
      rating string //rating on scale
      timemodified int //time modified (timestamp)
    }
  )warnings Optional //list of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}

```

XML-RPC (PHP structure)

```

Array
(
  [
    [ratings] =>
    Array
      (
        [
          [0] =>
          Array
            (
              [
                [id] => int
                [userid] => int
                [userpictureurl] => string
                [userfullname] => string
                [rating] => string
                [timemodified] => int
              )
            )
        )
    [warnings] =>
    Array
      (
        [
          [0] =>
          Array
            (
              [
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
              )
            )
        )
      )
  )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="ratings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userpictureurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userfullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rating">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_reportbuilder_audiences_delete**Delete audience from report**

Arguments

reportid (Required)

Report id

General structure

```
int //Report id
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

instanceid (Required)

Audience instance id

General structure

```
int //Audience instance id
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

Response

General structure

```
int
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_add

Add column to report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

uniqueidentifier (Required)

Unique identifier of the column

General structure

```
string //Unique identifier of the column
```

XML-RPC (PHP structure)

```
[uniqueidentifier] => string
```

REST (POST parameters)

```
uniqueidentifier= string
```

Response

General structure

```
object {
    hassortablecolumns int    //hassortablecolumns
    sortablecolumns   //sortablecolumns
    list of (
        object {
            id int    //id
            title string //title
            heading string //heading
            sortdirection int //sortdirection
            sortenabled int //sortenabled
            sortorder int //sortorder
            sorticon object {
                key string //key
                component string //component
                title string //title
            }
            movetitle string //movetitle
            sortenabledtitle string //sortenabledtitle
        }
        )helpicon string //helpicon
}
```

XML-RPC (PHP structure)

```
Array
(
    [hassortablecolumns] => int
    [sortablecolumns] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [title] => string
                    [heading] => string
                    [sortdirection] => int
                    [sortenabled] => int
                    [sortorder] => int
                    [sorticon] =>
                        Array
                        (
                            [key] => string
                            [component] => string
                            [title] => string
                        )
                    [movetitle] => string
                    [sortenabledtitle] => string
                )
            )
        )
    [helpicon] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hassortablecolumns">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortablecolumns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortdirection">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortenabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sorticon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortenabledtitle">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_delete

Delete column from report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

columnid (Required)

Column ID

General structure

```
int //Column ID
```

XML-RPC (PHP structure)

```
[columnid] => int
```

REST (POST parameters)

```
columnid= int
```

Response**General structure**

```
object {
  hassortablecolumns int //hassortablecolumns
  sortablecolumns //sortablecolumns
  list of (
    object {
      id int //id
      title string //title
      heading string //heading
      sortdirection int //sortdirection
      sortenabled int //sortenabled
      sortorder int //sortorder
      sorticon object {
        key string //key
        component string //component
        title string //title
      }
      movetitle string //movetitle
      sortenabledtitle string //sortenabledtitle
    }
    )helpicon string //helpicon
}
```

XML-RPC (PHP structure)

```
Array
(
    [hashtablecolumns] => int
    [sortablecolumns] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [title] => string
                        [heading] => string
                        [sortdirection] => int
                        [sortenabled] => int
                        [sortorder] => int
                        [sorticon] =>
                            Array
                            (
                                [
                                    [key] => string
                                    [component] => string
                                    [title] => string
                                )
                                [
                                    [movetitle] => string
                                    [sortenabledtitle] => string
                                )
                            )
                )
            )
        )
    [helpicon] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hassortablecolumns">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortablecolumns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortdirection">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortenabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sorticon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortenabledtitle">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_reorder

Re-order column within report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

columnid (Required)

Column ID

General structure

```
int //Column ID
```

XML-RPC (PHP structure)

```
[columnid] => int
```

REST (POST parameters)

```
columnid= int
```

position (Required)

New column position

General structure

```
int //New column position
```

XML-RPC (PHP structure)

```
[position] => int
```

REST (POST parameters)

```
position= int
```

Response

Success

General structure

```
int //Success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_sort_get

Retrieve column sorting for report

Arguments

reportid (Required)

Report ID

**General structure**

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

Response

General structure

```
object {
    hassortablecolumns int    //hassortablecolumns
    sortablecolumns   //sortablecolumns
    list of (
        object {
            id int    //id
            title string //title
            heading string //heading
            sortdirection int //sortdirection
            sortenabled int //sortenabled
            sortorder int //sortorder
            sorticon object {
                key string //key
                component string //component
                title string //title
            }
            movetitle string //movetitle
            sortenabledtitle string //sortenabledtitle
        }
        )helpicon string //helpicon
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [hassortablecolumns] => int
    [sortablecolumns] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [title] => string
                    [heading] => string
                    [sortdirection] => int
                    [sortenabled] => int
                    [sortorder] => int
                    [sorticon] =>
                        Array
                        (
                            [key] => string
                            [component] => string
                            [title] => string
                        )
                    [movetitle] => string
                    [sortenabledtitle] => string
                )
            )
        [helpicon] => string
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hassortablecolumns">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortablecolumns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortdirection">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortenabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sorticon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortenabledtitle">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_sort_reorder

Re-order column sorting within report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

columnid (Required)

Column ID

General structure

```
int //Column ID
```

XML-RPC (PHP structure)

```
[columnid] => int
```

REST (POST parameters)

```
columnid= int
```

position (Required)

New column sort position

General structure

```
int //New column sort position
```

XML-RPC (PHP structure)

```
[position] => int
```

REST (POST parameters)

```
position= int
```

Response

General structure

```
object {
    hassortablecolumns int    //hassortablecolumns
    sortablecolumns   //sortablecolumns
    list of (
        object {
            id int    //id
            title string //title
            heading string //heading
            sortdirection int //sortdirection
            sortenabled int //sortenabled
            sortorder int //sortorder
            sorticon object {
                key string //key
                component string //component
                title string //title
            }
            movetitle string //movetitle
            sortenabledtitle string //sortenabledtitle
        }
        )helpicon string //helpicon
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [hassortablecolumns] => int
    [sortablecolumns] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [title] => string
                    [heading] => string
                    [sortdirection] => int
                    [sortenabled] => int
                    [sortorder] => int
                    [sorticon] =>
                        Array
                        (
                            [key] => string
                            [component] => string
                            [title] => string
                        )
                    [movetitle] => string
                    [sortenabledtitle] => string
                )
            )
        [helpicon] => string
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hassortablecolumns">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortablecolumns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortdirection">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortenabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sorticon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortenabledtitle">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_columns_sort_toggle

Toggle sorting of column within report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

columnid (Required)

Column ID

General structure

```
int //Column ID
```

XML-RPC (PHP structure)

```
[columnid] => int
```

REST (POST parameters)

```
columnid= int
```

enabled (Required)

Sort enabled

General structure

```
int //Sort enabled
```

XML-RPC (PHP structure)

```
[enabled] => int
```

REST (POST parameters)

```
enabled= int
```

direction (Default to "4")

Sort direction

General structure

```
int Default to "4" //Sort direction
```

XML-RPC (PHP structure)

```
[direction] => int
```

REST (POST parameters)

```
direction= int
```

Response**General structure**

```
object {
    hassortablecolumns int //hassortablecolumns
    sortablecolumns //sortablecolumns
    list of (
        object {
            id int //id
            title string //title
            heading string //heading
            sortdirection int //sortdirection
            sortenabled int //sortenabled
            sortorder int //sortorder
            sorticon object {
                key string //key
                component string //component
                title string //title
            }
            movetitle string //movetitle
            sortenabledtitle string //sortenabledtitle
        }
        )helpicon string //helpicon
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [hassortablecolumns] => int
    [sortablecolumns] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [title] => string
                        [heading] => string
                        [sortdirection] => int
                        [sortenabled] => int
                        [sortorder] => int
                        [sorticon] =>
                            Array
                            (
                                [
                                    [key] => string
                                    [component] => string
                                    [title] => string
                                ]
                            )
                        [movetitle] => string
                        [sortenabledtitle] => string
                    )
                )
            )
        )
    [helpicon] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hassortablecolumns">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="sortablecolumns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="title">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortdirection">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortenabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sorticon">
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="component">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="title">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortenabledtitle">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_conditions_add

Add condition to report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

uniqueidentifier (Required)

Unique identifier of the condition

General structure

```
string //Unique identifier of the condition
```

XML-RPC (PHP structure)

```
[uniqueidentifier] => string
```

REST (POST parameters)

```
uniqueidentifier= string
```

Response

General structure

```
object {
hasavailableconditions int //hasavailableconditions
availableconditions //availableconditions
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
})
})
)
)hasactiveconditions int //hasactiveconditions
activeconditionsform string //activeconditionsform
helpicon string //helpicon
javascript string Optional //javascript
}
```

XML-RPC (PHP structure)

```
Array
(
    [hasavailableconditions] => int
    [availableconditions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [optiongroup] =>
                                Array
                                    (
                                        [text] => string
                                        [values] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [value] => string
                                                                [visiblename] => string
                                                            )
                                                )
                                        )
                                    )
                )
            )
)
[hasactiveconditions] => int
[activeconditionsform] => string
[helpicon] => string
[javascript] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailableconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availableconditions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
    <KEY name="hasactiveconditions">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="activeconditionsform">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="helpicon">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="javascript">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_conditions_delete

Delete condition from report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

conditionid (Required)

Condition ID

General structure

```
int //Condition ID
```

XML-RPC (PHP structure)

```
[conditionid] => int
```

REST (POST parameters)

```
conditionid= int
```

Response**General structure**

```
object {
hasavailableconditions int //hasavailableconditions
availableconditions //availableconditions
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
}
)
}
)
hasactiveconditions int //hasactiveconditions
activeconditionsform string //activeconditionsform
helpicon string //helpicon
javascript string Optional //javascript
}
```

XML-RPC (PHP structure)

```
Array
(
    [hasavailableconditions] => int
    [availableconditions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [optiongroup] =>
                                Array
                                    (
                                        [text] => string
                                        [values] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [value] => string
                                                                [visiblename] => string
                                                            )
                                                )
                                        )
                                    )
                )
            )
)
[hasactiveconditions] => int
[activeconditionsform] => string
[helpicon] => string
[javascript] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailableconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availableconditions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
    <KEY name="hasactiveconditions">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="activeconditionsform">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="helpicon">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="javascript">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_conditions_reorder**Re-order condition within report**

Arguments**reportid** (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

conditionid (Required)

Condition ID

General structure

```
int //Condition ID
```

XML-RPC (PHP structure)

```
[conditionid] => int
```

REST (POST parameters)

```
conditionid= int
```

position (Required)

New condition position

General structure

```
int //New condition position
```

XML-RPC (PHP structure)

```
[position] => int
```

**REST (POST parameters)**

```
position= int
```

Response

General structure

```
object {
hasavailableconditions int //hasavailableconditions
availableconditions //availableconditions
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
})
})
}
)hasactiveconditions int //hasactiveconditions
activeconditionsform string //activeconditionsform
helpicon string //helpicon
javascript string Optional //javascript
}
```

XML-RPC (PHP structure)

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailableconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availableconditions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
    <KEY name="hasactiveconditions">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="activeconditionsform">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="helpicon">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="javascript">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_conditions_reset

Reset conditions for given report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

Response

General structure

```
object {
    hasavailableconditions int //hasavailableconditions
    availableconditions //availableconditions
    list of (
        object {
            optiongroup object {
                text string //text
                values //values
            }
            list of (
                object {
                    value string //value
                    visiblename string //visiblename
                }
            )
        }
    )
    hasactiveconditions int //hasactiveconditions
    activeconditionsform string //activeconditionsform
    helpicon string //helpicon
    javascript string Optional //javascript
}
```



XML-RPC (PHP structure)

```
Array
(
    [hasavailableconditions] => int
    [availableconditions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [optiongroup] =>
                                Array
                                    (
                                        [text] => string
                                        [values] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [value] => string
                                                                [visiblename] => string
                                                            )
                                                )
                                        )
                                    )
                )
            )
)
[hasactiveconditions] => int
[activeconditionsform] => string
[helpicon] => string
[javascript] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailableconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availableconditions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="hasactiveconditions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="activeconditionsform">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="javascript">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_filters_add

Add filter to report

Arguments**reportid** (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

uniqueidentifier (Required)

Unique identifier of the filter

General structure

```
string //Unique identifier of the filter
```

XML-RPC (PHP structure)

```
[uniqueidentifier] => string
```

REST (POST parameters)

```
uniqueidentifier= string
```

Response**General structure**

```
object {
hasavailablefilters int //hasavailablefilters
availablefilters //availablefilters
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
})
})
)
)
hasactivefilters int //hasactivefilters
activefilters //activefilters
list of (
object {
id int //id
heading string //heading
headingeditable string //headingeditable
sortorder int //sortorder
movetitle string //movetitle
entityname string //entityname
})
)
helpicon string //helpicon
}
```

XML-RPC (PHP structure)

```
Array
(
    [hasavailablefilters] => int
    [availablefilters] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [optiongroup] =>
                                Array
                                    (
                                        [text] => string
                                        [values] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [value] => string
                                                                [visiblename] => string
                                                            )
                                                )
                                        )
                                    )
                )
            )
)
[hasactivefilters] => int
[activefilters] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [heading] => string
                        [headingeditable] => string
                        [sortorder] => int
                        [movetitle] => string
                        [entityname] => string
                    )
            )
)
[helpicon] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailablefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availablefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="hasactivefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="activefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="headingeditable">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="entityname">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_filters_delete

Delete filter from report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

filterid (Required)

Filter ID

General structure

```
int //Filter ID
```

XML-RPC (PHP structure)

```
[filterid] => int
```

REST (POST parameters)

```
filterid= int
```

Response

General structure

```

object {
hasavailablefilters int //hasavailablefilters
availablefilters //availablefilters
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
}
)
})
)
)hasactivefilters int //hasactivefilters
activefilters //activefilters
list of (
object {
id int //id
heading string //heading
headingeditable string //headingeditable
sortorder int //sortorder
movetitle string //movetitle
entityname string //entityname
}
)
)helpicon string //helpicon
}

```

XML-RPC (PHP structure)

```

Array
(
[hasavailablefilters] => int
[availablefilters] =>
    Array
        (
            [0] =>
                Array
                    (
                        [optiongroup] =>
                            Array
                                (
                                    [text] => string
                                    [values] =>
                                        Array
                                            (
                                                [0] =>
                                                    Array
                                                        (
                                                            [value] => string
                                                            [visiblename] => string
                                                        )
                                                    )
                                            )
                    )
            )
        )
)
[hasactivefilters] => int
[activefilters] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [heading] => string
                        [headingeditable] => string
                        [sortorder] => int
                        [movetitle] => string
                        [entityname] => string
                    )
            )
)
[helpicon] => string
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailablefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availablefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="hasactivefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="activefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="headingeditable">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="entityname">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_filters_reorder

Re-order filter within report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

filterid (Required)

Filter ID

General structure

```
int //Filter ID
```

XML-RPC (PHP structure)

```
[filterid] => int
```



REST (POST parameters)

```
filterid= int
```

position (Required)

New filter position

General structure

```
int //New filter position
```

XML-RPC (PHP structure)

```
[position] => int
```

REST (POST parameters)

```
position= int
```

Response

```
General structure

object {
hasavailablefilters int    //hasavailablefilters
availablefilters   //availablefilters
list of (
object {
optiongroup object {
text string   //text
values   //values
list of (
object {
value string   //value
visiblename string   //visiblename
}
)
}
)
hasactivefilters int    //hasactivefilters
activefilters   //activefilters
list of (
object {
id int   //id
heading string   //heading
headingeditable string   //headingeditable
sortorder int   //sortorder
movetitle string   //movetitle
entityname string   //entityname
}
)
helpicon string   //helpicon
}
```



XML-RPC (PHP structure)

```
Array
(
    [hasavailablefilters] => int
    [availablefilters] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [optiongroup] =>
                                Array
                                    (
                                        [text] => string
                                        [values] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [value] => string
                                                                [visiblename] => string
                                                            )
                                                )
                                        )
                                    )
                )
            )
)
[hasactivefilters] => int
[activefilters] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [heading] => string
                        [headingeditable] => string
                        [sortorder] => int
                        [movetitle] => string
                        [entityname] => string
                    )
            )
)
[helpicon] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="hasavailablefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="availablefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="optiongroup">
                        <SINGLE>
                            <KEY name="text">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="values">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="value">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="visiblename">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="hasactivefilters">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="activefilters">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="heading">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="headingeditable">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="movetitle">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="entityname">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_filters_reset

Reset filters for given report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

parameters (Default to "")

JSON encoded report parameters

General structure

```
string Default to "" //JSON encoded report parameters
```

XML-RPC (PHP structure)

```
[parameters] => string
```

REST (POST parameters)

```
parameters= string
```

Response

Success

General structure

```
int //Success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_list_reports

List custom reports for current user

Arguments**page** (Default to "0")

Page number

General structure

```
int Default to "0" //Page number
```

XML-RPC (PHP structure)

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

perpage (Default to "10")

Reports per page

General structure

```
int Default to "10" //Reports per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

Response

```
General structure

object {
reports list of (
object {
name string //name
source string //source
type int //type
uniquerows int //uniquerows
conditiondata string //conditiondata
settingsdata string //settingsdata
contextid int //contextid
component string //component
area string //area
itemid int //itemid
usercreated int //usercreated
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
sourcename string //sourcename
modifiedby object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
)
)
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})
```



XML-RPC (PHP structure)

```
Array
(
    [reports] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [source] => string
                            [type] => int
                            [uniquerows] => int
                            [conditiondata] => string
                            [settingsdata] => string
                            [contextid] => int
                            [component] => string
                            [area] => string
                            [itemid] => int
                            [usercreated] => int
                            [id] => int
                            [timecreated] => int
                            [timemodified] => int
                            [usermodified] => int
                            [sourcename] => string
                            [modifiedby] =>
                                Array
                                    (
                                        [id] => int
                                        [email] => string
                                        [idnumber] => string
                                        [phone1] => string
                                        [phone2] => string
                                        [department] => string
                                        [institution] => string
                                        [fullname] => string
                                        [identity] => string
                                        [profileurl] => string
                                        [profileimageurl] => string
                                        [profileimageurlsmall] => string
                                    )
                            )
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="reports">
      <MULTIPLE>
        <SINGLE>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="source">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="type">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="uniquerows">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="conditiondata">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="settingsdata">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="contextid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="component">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="area">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usercreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="usermodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sourcename">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="modifiedby">
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="email">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="idnumber">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="phone1">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="phone2">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="department">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="institution">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="fullname">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="identity">
```



```

<VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_reportbuilder_reports_delete

Delete report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

Response

Success

General structure

```
int //Success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_reportbuilder_reports_get

Get custom report

Arguments**reportid** (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

editmode (Default to "0")

Whether editing mode is enabled

General structure

```
int Default to "0" //Whether editing mode is enabled
```

XML-RPC (PHP structure)

```
[editmode] => int
```

REST (POST parameters)

```
editmode= int
```

pagesize (Default to "0")

Page size

General structure

```
int Default to "0" //Page size
```

XML-RPC (PHP structure)

```
[pagesize] => int
```

REST (POST parameters)

```
pagesize= int
```

**Response**

```

General structure

object {
name string //name
source string //source
type int //type
uniquerows int //uniquerows
conditiondata string //conditiondata
settingsdata string //settingsdata
contextid int //contextid
component string //component
area string //area
itemid int //itemid
usercreated int //usercreated
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
table string //table
filtersapplied int //filtersapplied
filterspresent int //filterspresent
filtersform string //filtersform
attributes //attributes
list of (
object {
name string //name
value string //value
})
)classes string //classes
editmode int //editmode
sidebarmenucards object {
menucards Optional //menucards
list of (
object {
name string //name
key string //key
items Optional //items
list of (
object {
name string //name
identifier string //identifier
title string //title
action string //action
disabled int Optional //disabled
})
)}
)}
conditions object {
hasavailableconditions int //hasavailableconditions
availableconditions //availableconditions
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
})
)})
)
)
hasactiveconditions int //hasactiveconditions
activeconditionsform string //activeconditionsform
helpicon string //helpicon
javascript string Optional //javascript
)
filters object {
hasavailablefilters int //hasavailablefilters
availablefilters //availablefilters
list of (
object {
optiongroup object {
text string //text
values //values
list of (
object {
value string //value
visiblename string //visiblename
})
)})
)
)
hasactivefilters int //hasactivefilters
}

```

^

```
activefilters //activefilters
list of (
object {
id int //id
heading string //heading
headingeditable string //headingeditable
sortorder int //sortorder
movetitle string //movetitle
entityname string //entityname
}
)helpicon string //helpicon
}
sorting object {
hassortablecolumns int //hassortablecolumns
sortablecolumns //sortablecolumns
list of (
object {
id int //id
title string //title
heading string //heading
sortdirection int //sortdirection
sortenabled int //sortenabled
sortorder int //sortorder
sorticon object {
key string //key
component string //component
title string //title
}
movetitle string //movetitle
sortenabledtitle string //sortenabledtitle
}
)helpicon string //helpicon
}
cardview object {
form string //form
helpicon string //helpicon
}
javascript string //javascript
}
```



XML-RPC (PHP structure)

```

Array
(
    [name] => string
    [source] => string
    [type] => int
    [uniquerows] => int
    [conditiondata] => string
    [settingsdata] => string
    [contextid] => int
    [component] => string
    [area] => string
    [itemid] => int
    [usercreated] => int
    [id] => int
    [timecreated] => int
    [timemodified] => int
    [usermodified] => int
    [table] => string
    [filtersapplied] => int
    [filterspresent] => int
    [filtersform] => string
    [attributes] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [value] => string
                        )
            )
    [classes] => string
    [editmode] => int
    [sidebarmenucards] =>
        Array
            (
                [menucards] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [key] => string
                                        [items] =>
                                            Array
                                                (
                                                    [0] =>
                                                        Array
                                                            (
                                                                [name] => string
                                                                [identifier] => string
                                                                [title] => string
                                                                [action] => string
                                                                [disabled] => int
                                                            )
                                                )
                                    )
            )
    [conditions] =>
        Array
            (
                [hasavailableconditions] => int
                [availableconditions] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [optiongroup] =>
                                            Array
                                                (
                                                    [text] => string
                                                    [values] =>
                                                        Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                )
                                                            )
                                                )
                                    )
            )
)

```





```
    [helpicon] => string
)
[cardview] =>
  Array
  (
    [form] => string
    [helpicon] => string
  )
[javascript] => string
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="source">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="type">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="uniquerows">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="conditiondata">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="settingsdata">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="component">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="area">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="itemid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usercreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="table">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="filtersapplied">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="filterspresent">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="filtersform">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="attributes">
      <MULTIPLE>
        <SINGLE>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="value">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="classes">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="editmode">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="sidebarmenucards">
      <SINGLE>
        <KEY name="menucards">
```



```
<MULTIPLE>
  <SINGLE>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="key">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="items">
      <MULTIPLE>
        <SINGLE>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="identifier">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="title">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="action">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="disabled">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="conditions">
  <SINGLE>
    <KEY name="hasavailableconditions">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="availableconditions">
      <MULTIPLE>
        <SINGLE>
          <KEY name="optiongroup">
            <SINGLE>
              <KEY name="text">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="values">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="value">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="visiblename">
                      <VALUE>string</VALUE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </SINGLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="hasactiveconditions">
  <VALUE>int</VALUE>
</KEY>
<KEY name="activeconditionsform">
  <VALUE>string</VALUE>
</KEY>
<KEY name="helpicon">
  <VALUE>string</VALUE>
</KEY>
<KEY name="javascript">
  <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="filters">
  <SINGLE>
    <KEY name="hasavailablefilters">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
```



```
<KEY name="availablefilters">
  <MULTIPLE>
    <SINGLE>
      <KEY name="optiongroup">
        <SINGLE>
          <KEY name="text">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="values">
            <MULTIPLE>
              <SINGLE>
                <KEY name="value">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="visiblename">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </KEY>
    </MULTIPLE>
  </KEY>
<KEY name="hasactivefilters">
  <VALUE>int</VALUE>
</KEY>
<KEY name="activefilters">
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="heading">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="headingeditable">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="sortorder">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="movetitle">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="entityname">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</KEY>
<KEY name="helpicon">
  <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="sorting">
  <SINGLE>
    <KEY name="hassortablecolumns">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="sortablecolumns">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="title">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="heading">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="sortdirection">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortenabled">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="sortorder">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</KEY>
```



```

<KEY name="sorticon">
    <SINGLE>
        <KEY name="key">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="title">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="sortenabledtitle">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="helpicon">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="cardview">
    <SINGLE>
        <KEY name="form">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="helpicon">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="javascript">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_retrieve_report

Retrieve custom report content

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

page (Default to "0")

Page number

General structure

```
int Default to "0" //Page number
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "10")

Reports per page

General structure

```
int Default to "10" //Reports per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```



Response

General structure

```
object {
details object {
name string //name
source string //source
type int //type
uniquerows int //uniquerows
conditiondata string //conditiondata
settingsdata string //settingsdata
contextid int //contextid
component string //component
area string //area
itemid int //itemid
usercreated int //usercreated
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
sourcename string //sourcename
modifiedby object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
}
data object {
headers //headers
list of (
string //headers
)rows //rows
list of (
object {
columns //columns
list of (
string //columns
)}
)
)totalrowcount int //totalrowcount
}
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```

Array
(
    [details] =>
        Array
            (
                [name] => string
                [source] => string
                [type] => int
                [uniquerows] => int
                [conditiondata] => string
                [settingsdata] => string
                [contextid] => int
                [component] => string
                [area] => string
                [itemid] => int
                [usercreated] => int
                [id] => int
                [timecreated] => int
                [timemodified] => int
                [usermodified] => int
                [sourcename] => string
                [modifiedby] =>
                    Array
                        (
                            [id] => int
                            [email] => string
                            [idnumber] => string
                            [phone1] => string
                            [phone2] => string
                            [department] => string
                            [institution] => string
                            [fullname] => string
                            [identity] => string
                            [profileurl] => string
                            [profileimageurl] => string
                            [profileimageurlsmall] => string
                        )
                )
            )
    [data] =>
        Array
            (
                [headers] =>
                    Array
                        (
                            [0] => string
                        )
                [rows] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [columns] =>
                                            Array
                                                (
                                                    [0] => string
                                                )
                                    )
                            )
                [totalrowcount] => int
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="details">
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="source">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="type">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="uniquerows">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="conditiondata">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="settingsdata">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="component">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="area">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usercreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="usermodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="sourcename">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="modifiedby">
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="email">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="idnumber">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="phone1">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="phone2">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="department">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="institution">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="identity">
            <VALUE>string</VALUE>
          </KEY>
```



```

        </KEY>
        <KEY name="profileurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimageurlsmall">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="data">
    <SINGLE>
        <KEY name="headers">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="rows">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="columns">
                        <MULTIPLE>
                            <VALUE>string</VALUE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="totalrowcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_reportbuilder_schedules_delete

Delete schedule from report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

scheduleid (Required)

Schedule ID

General structure

```
int //Schedule ID
```

XML-RPC (PHP structure)

```
[scheduleid] => int
```

REST (POST parameters)

```
scheduleid= int
```



Response

General structure

```
int
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_schedules_send

Send report schedule

Arguments**reportid** (Required)

Report ID

General structure

int //Report ID

XML-RPC (PHP structure)

[reportid] => int

REST (POST parameters)

reportid= int

scheduleid (Required)

Schedule ID

General structure

int //Schedule ID

**XML-RPC (PHP structure)**

[scheduleid] => int

REST (POST parameters)

scheduleid= int

Response**General structure**

int

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_schedules_toggle

Toggle state of report schedule

Arguments**reportid** (Required)

Report ID

General structure

```
int //Report ID
```

**XML-RPC (PHP structure)**

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

scheduleid (Required)

Schedule ID

General structure

```
int //Schedule ID
```

XML-RPC (PHP structure)

```
[scheduleid] => int
```

REST (POST parameters)

```
scheduleid= int
```

enabled (Required)

Schedule enabled

General structure

```
int //Schedule enabled
```

XML-RPC (PHP structure)

```
[enabled] => int
```

REST (POST parameters)

```
enabled= int
```

Response**General structure**

```
int
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_set_filters

Set filter values for given report

Arguments

reportid (Required)

Report ID

General structure

```
int //Report ID
```

XML-RPC (PHP structure)

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

parameters (Default to "")

JSON encoded report parameters

General structure

```
string Default to "" //JSON encoded report parameters
```

XML-RPC (PHP structure)

```
[parameters] => string
```

REST (POST parameters)

```
parameters= string
```

values (Required)

JSON encoded filter values

General structure

```
string //JSON encoded filter values
```



XML-RPC (PHP structure)

```
[values] => string
```

REST (POST parameters)

```
values= string
```

Response

Success

General structure

```
int //Success
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_reportbuilder_view_report

Trigger custom report viewed

Arguments**reportid** (Required)

Report ID

General structure

```
int //Report ID
```

**XML-RPC (PHP structure)**

```
[reportid] => int
```

REST (POST parameters)

```
reportid= int
```

Response

General structure

```

object {
    status int //Success
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`core_role_assign_roles`

Manual role assignments.

Arguments

`assignments` (Required)

General structure

```
list of (
object {
roleid int //Role to assign to the user
userid int //The user that is going to be assigned
contextid int Optional //The context to assign the user role in
contextlevel string Optional //The context level to assign the user role in
(instanceid int Optional //The Instance id of item where the role needs to be assigned
)
}
```

XML-RPC (PHP structure)

```
[assignments] =>
Array
(
[0] =>
Array
(
[roleid] => int
[userid] => int
[contextid] => int
[contextlevel] => string
[instanceid] => int
)
)
```

REST (POST parameters)

```
assignments[0][roleid]= int
assignments[0][userid]= int
assignments[0][contextid]= int
assignments[0][contextlevel]= string
assignments[0][instanceid]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_role_unassign_roles

Manual role unassignments.

Arguments**unassignments** (Required)**General structure**

```
list of (
object {
roleid int //Role to assign to the user
userid int //The user that is going to be assigned
contextid int Optional //The context to unassign the user role from
contextlevel string Optional //The context level to unassign the user role in
+                               (block, course, coursecat, system, user, module)
instanceid int Optional //The Instance id of item where the role needs to be unassigned
})
```

XML-RPC (PHP structure)

```
[unassignments] =>
    Array
        (
            [0] =>
                Array
                    (
                        [
                            [roleid] => int
                            [userid] => int
                            [contextid] => int
                            [contextlevel] => string
                            [instanceid] => int
                        )
        )
```

**REST (POST parameters)**

```
unassignments[0][roleid]= int
unassignments[0][userid]= int
unassignments[0][contextid]= int
unassignments[0][contextlevel]= string
unassignments[0][instanceid]= int
```

Response**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`core_search_get_relevant_users`

Gets relevant users for a search request.

Arguments

`query` (Required)

Query string (full or partial user full name or other details)

General structure

```
string //Query string (full or partial user full name or other details)
```

XML-RPC (PHP structure)

```
[query] => string
```

REST (POST parameters)

```
query= string
```

`courseid` (Required)

Course id (0 if none)

General structure

```
int //Course id (0 if none)
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
list of (
object {
id int    //User id
fullname string  //Full name as text
profileimageurlsmall string  //URL to small profile image
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[id] => int
[fullname] => string
[profileimageurlsmall] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="fullname">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_session_time_remaining

Count the seconds remaining in this session

Arguments

Response**General structure**

```
object {
    userid int //The current user id.
    timeremaining int //The number of seconds remaining in this session.
}
```

XML-RPC (PHP structure)

```
Array
(
    [userid] => int
    [timeremaining] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeremaining">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_session_touch

Keep the users session alive

Arguments**Response**

result

General structure

```
int //result
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_table_get_dynamic_table_content

Get the dynamic table content raw html

Arguments

component (Required)

Component

**General structure**

```
string //Component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

handler (Required)

Handler

General structure

```
string //Handler
```

XML-RPC (PHP structure)

```
[handler] => string
```

REST (POST parameters)

```
handler= string
```

uniqueid (Required)

Unique ID for the container

General structure

```
string //Unique ID for the container
```

XML-RPC (PHP structure)

```
[uniqueid] => string
```

REST (POST parameters)

```
uniqueid= string
```

sortdata (Optional)

The combined sort order of the table. Multiple fields can be specified.

General structure

```
Optional //The combined sort order of the table. Multiple fields can be specified.  
list of (  
object {  
sortby string //The name of a sortable column  
sortorder string //The direction that this column should be sorted by  
})  
)
```

XML-RPC (PHP structure)

```
[sortdata] =>  
Array  
(  
[0] =>  
Array  
(  
[sortby] => string  
[sortorder] => string  
))
```

**REST (POST parameters)**

```
sortdata[0][sortby]= string  
sortdata[0][sortorder]= string
```

filters (Optional)

The filters that will be applied in the request

General structure

```
Optional //The filters that will be applied in the request
list of (
object {
name string //Name of the filter
jointype int //Type of join for filter values
values //The value to filter on
list of (
string //Filter value
)})
)
```

XML-RPC (PHP structure)

```
[filters] =>
Array
(
[0] =>
Array
(
[name] => string
[jointype] => int
[values] =>
Array
(
[0] => string
)
)
)
```

REST (POST parameters)

```
filters[0][name]= string
filters[0][jointype]= int
filters[0][values][0]= string
```

jointype (Required)

Type of join to join all filters together

General structure

```
int //Type of join to join all filters together
```

XML-RPC (PHP structure)

```
[jointype] => int
```

REST (POST parameters)

```
jointype= int
```

firstinitial (Required)

The first initial to sort filter on

General structure

```
string //The first initial to sort filter on
```

XML-RPC (PHP structure)

```
[firstinitial] => string
```

REST (POST parameters)

```
firstinitial= string
```

lastinitial (Required)

The last initial to sort filter on

General structure

```
string //The last initial to sort filter on
```

XML-RPC (PHP structure)

```
[lastinitial] => string
```

REST (POST parameters)

```
lastinitial= string
```

pagenumber (Required)

The page number

General structure

```
int //The page number
```

XML-RPC (PHP structure)

```
[pagenumber] => int
```

REST (POST parameters)

```
pagenumber= int
```

pagesize (Required)

The number of records per page

General structure

```
int //The number of records per page
```

XML-RPC (PHP structure)

```
[pagesize] => int
```

REST (POST parameters)

```
pagesize= int
```

hiddencolumns (Required)**General structure**

```
list of (
string //Name of column
)
```

XML-RPC (PHP structure)

```
[hiddencolumns] =>
Array
(
    [
        [0] => string
    ]
)
```

REST (POST parameters)

```
hiddencolumns[0]= string
```

resetpreferences (Required)

Whether the table preferences should be reset

General structure

```
int //Whether the table preferences should be reset
```

XML-RPC (PHP structure)

```
[resetpreferences] => int
```

REST (POST parameters)

```
resetpreferences= int
```

Response**General structure**

```
object {
    html string //The raw html of the requested table.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [html] => string
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="html">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

**core_tag_get_tag_areas**

Retrieves existing tag areas.

Arguments**Response**

General structure

```
object {
areas list of (
object {
id int //Area id.
component string //Component the area is related to.
itemtype string //Type of item in the component.
enabled int //Whether this area is enabled.
tagcollid int //The tag collection this area belongs to.
callback string //Component callback for processing tags.
callbackfile string //Component callback file.
showstandard int //Return whether to display only standard, only non-standard or both tags.
multiplecontexts int //Whether the tag area allows tag instances to be created in multiple contexts.
locked int Optional //Whether the area is Locked.
}
)warnings Optional //list of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

XML-RPC (PHP structure)

```
Array
(
    [areas] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [component] => string
                    [itemtype] => string
                    [enabled] => int
                    [tagcollid] => int
                    [callback] => string
                    [callbackfile] => string
                    [showstandard] => int
                    [multiplecontexts] => int
                    [locked] => int
                )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="areas">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="enabled">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="tagcollid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="callback">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="callbackfile">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="showstandard">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="multiplecontexts">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="locked">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_tag_get_tag_cloud

Retrieves a tag cloud for the given collection and/or query search.

Arguments

tagcollid (Default to "0")

Tag collection id.

General structure

```
int Default to "0" //Tag collection id.
```

XML-RPC (PHP structure)

```
[tagcollid] => int
```

REST (POST parameters)

```
tagcollid= int
```

isstandard (Default to "")

Whether to return only standard tags.

General structure

```
int Default to "" //Whether to return only standard tags.
```

XML-RPC (PHP structure)

```
[isstandard] => int
```

REST (POST parameters)

```
isstandard= int
```



limit (Default to "150")

Maximum number of tags to retrieve.

General structure

```
int Default to "150" //Maximum number of tags to retrieve.
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

sort (Default to "name")

Sort order for display (id, name, rawname, count, flag, isstandard, tagcollid).

General structure

```
string Default to "name" //Sort order for display
        (id, name, rawname, count, flag, isstandard, tagcollid).
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

search (Default to "")

Search string.

General structure

```
string Default to "" //Search string.
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

fromctx (Default to "0")

Context id where this tag cloud is displayed.

General structure

```
int Default to "0" //Context id where this tag cloud is displayed.
```

**XML-RPC (PHP structure)**

```
[fromctx] => int
```

REST (POST parameters)

```
fromctx= int
```

ctx (Default to "0")

Only retrieve tag instances in this context.

General structure

```
int Default to "0" //Only retrieve tag instances in this context.
```

XML-RPC (PHP structure)

```
[ctx] => int
```

REST (POST parameters)

```
ctx= int
```

rec (Default to "1")

Retrieve tag instances in the \$ctx context and it's children.

General structure

```
int Default to "1" //Retrieve tag instances in the $ctx context and it's children.
```

XML-RPC (PHP structure)

```
[rec] => int
```

REST (POST parameters)

```
rec= int
```

Response**General structure**

```
object {
tags list of (
    //Tags.
object {
name string //Tag name.
viewurl string //URL to view the tag index.
flag int Optional //Whether the tag is flagged as inappropriate.
isstandard int Optional //Whether is a standard tag or not.
count int Optional //Number of tag instances.
size int Optional //Proportional size to display the tag.
}
)tagscount int //Number of tags returned.
totalcount int //Total count of tags.
warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}
```



XML-RPC (PHP structure)

```
Array
(
    [tags] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [viewurl] => string
                            [flag] => int
                            [isstandard] => int
                            [count] => int
                            [size] => int
                        )
                )
            )
    [tagscount] => int
    [totalcount] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="tags">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="viewurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="flag">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isstandard">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="count">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="size">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="tagscount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="totalcount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_tag_get_tag_collections

Retrieves existing tag collections.

Arguments**Response**

General structure

```
object {
  collections list of (
    object {
      id int //Collection id.
      name string //Collection name.
      isdefault int //Whether is the default collection.
      component string //Component the collection is related to.
      sortorder int //Collection ordering in the list.
      searchable int //Whether the tag collection is searchable.
      customurl string //Custom URL for the tag page instead of /tag/index.php.
    }
    )warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )}
```

XML-RPC (PHP structure)

```
Array
(
  [
    [collections] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [id] => int
                    [name] => string
                    [isdefault] => int
                    [component] => string
                    [sortorder] => int
                    [searchable] => int
                    [customurl] => string
                  )
                )
            ]
        )
    [
      [warnings] =>
        Array
          (
            [
              [0] =>
                Array
                  (
                    [
                      [item] => string
                      [itemid] => int
                      [warningcode] => string
                      [message] => string
                    )
            )
        )
    )
  )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="collections">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isdefault">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="sortorder">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="searchable">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="customurl">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_tag_get_tagindex

Gets tag index page for one tag and one tag area

Arguments

tagindex (Required)

parameters

General structure

```
//parameters
object {
    tag string //tag name
    tc int //tag collection id
    ta int //tag area id
    excl int Optional //exclusive mode for this tag area
    from int Optional //context id where the link was displayed
    ctx int Optional //context id where to search for items
    rec int Optional //search in the context recursive
    page int Optional //page number (0-based)
}
```

XML-RPC (PHP structure)

```
[tagindex] =>
    Array
        (
            [tag] => string
            [tc] => int
            [ta] => int
            [excl] => int
            [from] => int
            [ctx] => int
            [rec] => int
            [page] => int
        )
```

REST (POST parameters)

```
tagindex[tag]= string
tagindex[tc]= int
tagindex[ta]= int
tagindex[excl]= int
tagindex[from]= int
tagindex[ctx]= int
tagindex[rec]= int
tagindex[page]= int
```

Response

tag index

General structure

```
//tag index
object {
tagid int    //tag id
ta int     //tag area id
component string   //component
itemtype string   //itemtype
nextpageurl string  Optional //URL for the next page
prevpageurl string  Optional //URL for the next page
exclusiveurl string  Optional //URL for exclusive Link
exclusivetext string  Optional //text for exclusive Link
title string   //title
content string   //title
hascontent int   //whether the content is present
anchor string  Optional //name of anchor
}
```

XML-RPC (PHP structure)

```
Array
(
[tagid] => int
[ta] => int
[component] => string
[itemtype] => string
[nextpageurl] => string
[prevpageurl] => string
[exclusiveurl] => string
[exclusivetext] => string
[title] => string
[content] => string
[hascontent] => int
[anchor] => string
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="tagid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ta">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="component">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="itemtype">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="nextpageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="prevpageurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="exclusiveurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="exclusivetext">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="title">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="content">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="hascontent">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="anchor">
      <VALUE>string</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_tag_get_tagindex_per_area

Gets tag index page per different areas.

Arguments

tagindex (Required)

parameters

General structure

```
//parameters
object {
id int  Optional //tag id
tag string  Optional //tag name
tc int  Optional //tag collection id
ta int  Optional //tag area id
excl int  Optional //exclusive mode for this tag area
from int  Optional //context id where the Link was displayed
ctx int  Optional //context id where to search for items
rec int  Optional //search in the context recursive
page int  Optional //page number (0-based)
}
```

XML-RPC (PHP structure)

```
[tagindex] =>
Array
(
    [id] => int
    [tag] => string
    [tc] => int
    [ta] => int
    [excl] => int
    [from] => int
    [ctx] => int
    [rec] => int
    [page] => int
)
```

REST (POST parameters)

```
tagindex[id]= int
tagindex[tag]= string
tagindex[tc]= int
tagindex[ta]= int
tagindex[excl]= int
tagindex[from]= int
tagindex[ctx]= int
tagindex[rec]= int
tagindex[page]= int
```

Response

General structure

```
list of (
    //tag index
object {
tagid int  //tag id
ta int  //tag area id
component string  //component
itemtype string  //itemtype
nextpageurl string  Optional //URL for the next page
prevpageurl string  Optional //URL for the previous page
exclusiveurl string  Optional //URL for exclusive Link
exclusivetext string  Optional //text for exclusive Link
title string  //title
content string  //title
hascontent int  //whether the content is present
anchor string  Optional //name of anchor
})
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [tagid] => int
                [ta] => int
                [component] => string
                [itemtype] => string
                [nextpageurl] => string
                [prevpageurl] => string
                [exclusiveurl] => string
                [exclusivetext] => string
                [title] => string
                [content] => string
                [hascontent] => int
                [anchor] => string
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="tagid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="ta">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemtype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="nextpageurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="prevpageurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="exclusiveurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="exclusivetext">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="title">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="content">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="hascontent">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="anchor">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`core_tag_get_tags`

Gets tags by their ids

Arguments

`tags` (Required)

General structure

```
list of (
object {
id int //tag id
}
)
```

XML-RPC (PHP structure)

```
[tags] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                    )
        )
)
```

REST (POST parameters)

```
tags[0][id]= int
```

Response

General structure

```
object {
tags list of (
    //information about one tag
object {
id int //tag id
tagcollid int //tag collection id
name string //name
rawname string //tag raw name (may contain capital letters)
description string //tag description
descriptionformat int //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
flag int Optional //flag
official int Optional //whether this flag is standard (deprecated, use isstandard)
isstandard int Optional //whether this flag is standard
viewurl string //URL to view
}
)warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)})}
```

XML-RPC (PHP structure)

```

Array
(
    [tags] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [tagcollid] => int
                            [name] => string
                            [rawname] => string
                            [description] => string
                            [descriptionformat] => int
                            [flag] => int
                            [official] => int
                            [isstandard] => int
                            [viewurl] => string
                        )
                    )
            )
        )
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
        )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="tags">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="tagcollid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rawname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="flag">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="official">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isstandard">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="viewurl">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_tag_update_tags

Updates tags

Arguments

tags (Required)

General structure

```
list of (
object {
id int //tag id
rawname string Optional //tag raw name (may contain capital Letters)
description string Optional //tag description
descriptionformat int Optional //tag description format
flag int Optional //flag
official int Optional //deprecated, use isstandard) whether this flag is standard
isstandard int Optional //whether this flag is standard
}
)
```

XML-RPC (PHP structure)

```
[tags] =>
Array
(
[0] =>
Array
(
[id] => int
[rawname] => string
[description] => string
[descriptionformat] => int
[flag] => int
[official] => int
[isstandard] => int
)
)
```

REST (POST parameters)

```
tags[0][id]= int
tags[0][rawname]= string
tags[0][description]= string
tags[0][descriptionformat]= int
tags[0][flag]= int
tags[0][official]= int
tags[0][isstandard]= int
```



Response

General structure

```
object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
          )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_update_inplace_editable

Generic service to update title

Arguments

component (Required)

component responsible for the update

General structure

```
string //component responsible for the update
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

itemtype (Required)

type of the updated item inside the component

General structure

```
string //type of the updated item inside the component
```

XML-RPC (PHP structure)

```
[itemtype] => string
```

REST (POST parameters)

```
itemtype= string
```



itemid (Required)

identifier of the updated item

General structure

```
string //identifier of the updated item
```

XML-RPC (PHP structure)

```
[itemid] => string
```

REST (POST parameters)

```
itemid= string
```

value (Required)

new value

General structure

```
string //new value
```

XML-RPC (PHP structure)

```
[value] => string
```

REST (POST parameters)

```
value= string
```

Response**General structure**

```
object {
    displayvalue string //display value (may contain Link or other html tags)
    component string Optional //component responsible for the update
    itemtype string Optional //itemtype
    value string Optional //value of the item as it is stored
    itemid string Optional //identifier of the updated item
    edithint string Optional //hint for editing element
    editlabel string Optional //label for editing element
    editicon object {
        key string Optional //Edit icon key
        component string Optional //Edit icon component
        title string Optional //Edit icon title
    }
    type string Optional //type of the element (text, toggle, select)
    options string Optional //options of the element, format depends on type
    linkeverything int Optional //Should everything be wrapped in the edit link or Link displayed separately
}
```

XML-RPC (PHP structure)

```
Array
(
    [displayvalue] => string
    [component] => string
    [itemtype] => string
    [value] => string
    [itemid] => string
    [edithint] => string
    [editlabel] => string
    [editicon] =>
        Array
        (
            [
                [key] => string
                [component] => string
                [title] => string
            ]
        )
    [type] => string
    [options] => string
    [linkeverything] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="displayvalue">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemtype">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="value">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="edithint">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="editlabel">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="editicon">
            <SINGLE>
                <KEY name="key">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="component">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="title">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="type">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="options">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="linkeverything">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_user_add_user_device

Store mobile user devices information for PUSH Notifications.

Arguments

appid (Required)

the app id, usually something like com.moodle.moodlemobile

General structure

```
string //the app id, usually something like com.moodle.moodlemobile
```

XML-RPC (PHP structure)

```
[appid] => string
```

REST (POST parameters)

```
appid= string
```

name (Required)

the device name, 'occam' or 'iPhone' etc.

General structure

```
string //the device name, 'occam' or 'iPhone' etc.
```

XML-RPC (PHP structure)

```
[name] => string
```

REST (POST parameters)

```
name= string
```

model (Required)

the device model 'Nexus4' or 'iPad1,1' etc.

General structure

```
string //the device model 'Nexus4' or 'iPad1,1' etc.
```



XML-RPC (PHP structure)

```
[model] => string
```

REST (POST parameters)

```
model= string
```

platform (Required)

the device platform 'iOS' or 'Android' etc.

General structure

```
string //the device platform 'iOS' or 'Android' etc.
```

XML-RPC (PHP structure)

```
[platform] => string
```

REST (POST parameters)

```
platform= string
```

version (Required)

the device version '6.1.2' or '4.2.2' etc.

General structure

```
string //the device version '6.1.2' or '4.2.2' etc.
```

XML-RPC (PHP structure)

```
[version] => string
```

REST (POST parameters)

```
version= string
```

pushid (Required)

the device PUSH token/key/identifier/registration id

General structure

```
string //the device PUSH token/key/identifier/registration id
```

XML-RPC (PHP structure)

```
[pushid] => string
```

REST (POST parameters)

```
pushid= string
```

uuid (Required)

the device UUID

General structure

```
string //the device UUID
```

**XML-RPC (PHP structure)**

```
[uuid] => string
```

REST (POST parameters)

```
uuid= string
```

publickey (Default to "null")

the app generated public key

General structure

```
string Default to "null" //the app generated public key
```

XML-RPC (PHP structure)

```
[publickey] => string
```

REST (POST parameters)

```
publickey= string
```

Response**General structure**

```
list of (
  Optional //List of warnings
list of (
  //warning
object {
  item string Optional //item
  itemid int Optional //item id
  warningcode string //the warning code can be used by the client app to implement specific behaviour
  message string //untranslated english message to explain the warning
}
))
```

XML-RPC (PHP structure)

```
Array
(
  [
  [0] =>
    Array
    (
      [
      [0] =>
        Array
        (
          [
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
          )
        )
      )
    )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <MULTIPLE>
      <SINGLE>
        <KEY name="item">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="warningcode">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="message">
          <VALUE>string</VALUE>
        </KEY>
      </SINGLE>
    </MULTIPLE>
  </MULTIPLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_add_user_private_files

Copy files from a draft area to users private files area.

Arguments**draftid** (Required)

draft area id

General structure

```
int //draft area id
```

XML-RPC (PHP structure)

```
[draftid] => int
```

REST (POST parameters)

```
draftid= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_agree_site_policy

Agree the site policy for the current user.

Arguments

Response

```
General structure

object {
    status int   //Status: true only if we set the policyagreed to 1 for the user
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_create_users

Create users.

Arguments

users (Required)

General structure

```
list of (
object {
createpassword int  Optional //True if password should be created and mailed to user.
username string  //Username policy is defined in Moodle security config.
auth string  Default to "manual" //Auth plugins include manual, Ldap, etc
password string  Optional //Plain text password consisting of any characters
firstname string  //The first name(s) of the user
lastname string  //The family name of the user
email string  //A valid and unique email address
maildisplay int  Optional //Email visibility
city string  Optional //Home city of the user
country string  Optional //Home country code of the user, such as AU or CZ
timezone string  Optional //Timezone code such as Australia/Perth, or 99 for default
description string  Optional //User profile description, no HTML
firstnamephonetic string  Optional //The first name(s) phonetically of the user
lastnamephonetic string  Optional //The family name phonetically of the user
middlename string  Optional //The middle name of the user
alternatename string  Optional //The alternate name of the user
interests string  Optional //User interests (separated by commas)
idnumber string  Default to "" //An arbitrary ID code number perhaps from the institution
institution string  Optional //institution
department string  Optional //department
phone1 string  Optional //Phone 1
phone2 string  Optional //Phone 2
address string  Optional //Postal address
lang string  Default to "en" //Language code such as "en", must exist on server
calendartype string  Default to "gregorian" //Calendar type such as "gregorian", must exist on server
theme string  Optional //Theme name such as "standard", must exist on server
mailformat int  Optional //Mail format code is 0 for plain text, 1 for HTML etc
customfields object {
list of (
object {
type string  //The name of the custom field
value string  //The value of the custom field
})
preferences object {
list of (
object {
type string  //The name of the preference
value string  //The value of the preference
})
})}
```

XML-RPC (PHP structure)

```
[users] =>
    Array
        (
            [0] =>
                Array
                    (
                        [createpassword] => int
                        [username] => string
                        [auth] => string
                        [password] => string
                        [firstname] => string
                        [lastname] => string
                        [email] => string
                        [maildisplay] => int
                        [city] => string
                        [country] => string
                        [timezone] => string
                        [description] => string
                        [firstnamephonetic] => string
                        [lastnamephonetic] => string
                        [middlename] => string
                        [alternatename] => string
                        [interests] => string
                        [idnumber] => string
                        [institution] => string
                        [department] => string
                        [phone1] => string
                        [phone2] => string
                        [address] => string
                        [lang] => string
                        [calendartype] => string
                        [theme] => string
                        [mailformat] => int
                        [customfields] =>
                            Array
                                (
                                    [0] =>
                                        Array
                                            (
                                                [type] => string
                                                [value] => string
                                            )
                                )
                        )
                    )
            [preferences] =>
                Array
                    (
                        [0] =>
                            Array
                                (
                                    [type] => string
                                    [value] => string
                                )
                    )
        )
    )
)
```



```
REST (POST parameters)

users[0][createpassword]= int
users[0][username]= string
users[0][auth]= string
users[0][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][maildisplay]= int
users[0][city]= string
users[0][country]= string
users[0][timezone]= string
users[0][description]= string
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
users[0][interests]= string
users[0][idnumber]= string
users[0][institution]= string
users[0][department]= string
users[0][phone1]= string
users[0][phone2]= string
users[0][address]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][mailformat]= int
users[0][customfields][0][type]= string
users[0][customfields][0][value]= string
users[0][preferences][0][type]= string
users[0][preferences][0][value]= string
```

Response**General structure**

```
list of (
object {
id int    //user id
username string //user name
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
  Array
  (
    [id] => int
    [username] => string
  )
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_delete_users

Delete users.

Arguments**userids** (Required)**General structure**

```
list of (
int //user ID
)
```

XML-RPC (PHP structure)

```
[userids] =>
  Array
    (
      [0] => int
    )
```

REST (POST parameters)

userids[0]= int

**Response**

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`core_user_get_course_user_profiles`

Get course user profiles (each of the profils matching a course id and a user id).,

Arguments

`userlist` (Required)

General structure

```
list of (
object {
userid int  //userid
courseid int  //courseid
}
)
```

XML-RPC (PHP structure)

```
[userlist] =>
  Array
    (
      [0] =>
        Array
          (
            [userid] => int
            [courseid] => int
          )
    )
)
```

REST (POST parameters)

```
userlist[0][userid]= int
userlist[0][courseid]= int
```

Response



General structure

```

list of (
object {
id int //ID of the user
username string Optional //The username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //An email address - allow email as root@localhost
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //An arbitrary ID code number perhaps from the institution
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
auth string Optional //Auth plugins include manual, Ldap, etc
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
lang string Optional //Language code such as "en", must exist on server
calendartype string Optional //Calendar type such as "gregorian", must exist on server
theme string Optional //Theme name such as "standard", must exist on server
timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
description string Optional //User profile description
descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string //User image profile URL - small version
profileimageurl string //User image profile URL - big version
customfields Optional //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field (as stored in the database)
displayvalue string Optional //The value of the custom field for display
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
}
)preferences Optional //Users preferences
list of (
object {
name string //The name of the preferences
value string //The value of the preference
}
)groups Optional //user groups
list of (
object {
id int //group id
name string //group name
description string //group description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)roles Optional //user roles
list of (
object {
roleid int //role id
name string //role name
shortname string //role shortname
sortorder int //role sortorder
}
)enrolleddcourses Optional //Courses where the user is enrolled - Limited by which courses the user is able to see
list of (
object {
id int //Id of the course
fullname string //Fullname of the course
shortname string //Shortname of the course
}
)
})
)

```



XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
            (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [suspended] => int
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => string
                                        [value] => string
                                        [displayvalue] => string
                                        [name] => string
                                        [shortname] => string
                                    )
                        )
                [preferences] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [value] => string
                                    )
                        )
                [groups] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [id] => int
                                        [name] => string
                                        [description] => string
                                        [descriptionformat] => int
                                    )
                        )
                [roles] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [roleid] => int
                                        [name] => string
                                        [shortname] => string
                                        [sortorder] => int
                                    )
                        )
            )
)

```



```
        )
    [enrolledcourses] =>
    Array
    (
        [
            [0] =>
            Array
            (
                [
                    [id] => int
                    [fullname] => string
                    [shortname] => string
                )
            )
        )
    )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="auth">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="suspended">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="confirmed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="calendartype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="theme">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timezone">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="mailformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
```



```
</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="groups">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="roles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="roleid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="sortorder">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="enrolledcourses">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
```



```

        <VALUE>int</VALUE>
    </KEY>
    <KEY name="fullname">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="shortname">
        <VALUE>string</VALUE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_get_private_files_info

Returns general information about files in the user private files area.

Arguments

userid (Default to "0")

Id of the user, default to current user.

General structure

int *Default to "0" //Id of the user, default to current user.*

XML-RPC (PHP structure)

[userid] => int

REST (POST parameters)

userid= int

Response

General structure

```

object {
    filecount int //Number of files in the area.
    foldercount int //Number of folders in the area.
    filesize int //Total size of the files in the area.
    filesizewithoutreferences int //Total size of the area excluding file references
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [filecount] => int
    [foldercount] => int
    [filesize] => int
    [filesizewithoutreferences] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="filecount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="foldercount">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="filesize">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="filesizewithoutreferences">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_get_user_preferences

Return user preferences.

Arguments

name (Default to "")

preference name, empty for all

General structure

```
string Default to "" //preference name, empty for all
```

XML-RPC (PHP structure)

```
[name] => string
```

REST (POST parameters)

```
name= string
```

userid (Default to "0")

id of the user, default to current user



General structure

```
int Default to "0" //id of the user, default to current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

General structure

```
object {
preferences  //User custom fields (also known as user profile fields)
list of (
object {
name string //The name of the preference
value string //The value of the preference
})
warnings  Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}`
```

XML-RPC (PHP structure)

```
Array
(
[preferences] =>
    Array
        (
[0] =>
            Array
                (
[name] => string
[value] => string
)
)
[warnings] =>
    Array
        (
[0] =>
        Array
            (
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="preferences">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_user_get_users

search for users matching the parameters

Arguments**criteria (Required)**

the key/value pairs to be considered in user search. Values can not be empty. Specify different keys only once (fullname => 'user1', auth => 'manual', ...) - key occurrences are forbidden. The search is executed with AND operator on the criterias. Invalid criterias (keys) are ignored, the search is still executed on the valid criterias. You can search without criteria, but the function is not designed for it. It could be very slow or timeout. The function is designed to search some specific users.

General structure

```
//the key/value pairs to be considered in user search. Values can not be empty.
    Specify different keys only once (fullname => 'user1', auth => 'manual', ...) -
    key occurrences are forbidden.
    The search is executed with AND operator on the criterias. Invalid criterias (keys) are ignored,
    the search is still executed on the valid criterias.
    You can search without criteria, but the function is not designed for it.
    It could very slow or timeout. The function is designed to search some specific users.

list of
object {
key string //the user column to search, expected keys (value format) are:
        "id" (int) matching user id,
        "lastname" (string) user last name (Note: you can use % for searching but it may be considerably slower!),
        "firstname" (string) user first name (Note: you can use % for searching but it may be considerably slower!),
        "idnumber" (string) matching user idnumber,
        "username" (string) matching user username,
        "email" (string) user email (Note: you can use % for searching but it may be considerably slower!),
        "auth" (string) matching user auth plugin
value string //the value to search
}
}
```

XML-RPC (PHP structure)

```
[criteria] =>
    Array
        (
            [0] =>
                Array
                    (
                        [key] => string
                        [value] => string
                    )
        )
    )
```

REST (POST parameters)

```
criteria[0][key]= string
criteria[0][value]= string
```

Response

General structure

```

object {
  users list of (
    object {
      id int //ID of the user
      username string Optional //The username
      firstname string Optional //The first name(s) of the user
      lastname string Optional //The family name of the user
      fullname string //The fullname of the user
      email string Optional //An email address - allow email as root@localhost
      address string Optional //Postal address
      phone1 string Optional //Phone 1
      phone2 string Optional //Phone 2
      department string Optional //department
      institution string Optional //institution
      idnumber string Optional //An arbitrary ID code number perhaps from the institution
      interests string Optional //user interests (separated by commas)
      firstaccess int Optional //first access to the site (0 if never)
      lastaccess int Optional //last access to the site (0 if never)
      auth string Optional //Auth plugins include manual, Ldap, etc
      suspended int Optional //Suspend user account, either false to enable user login or true to disable it
      confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
      lang string Optional //Language code such as "en", must exist on server
      calendartype string Optional //Calendar type such as "gregorian", must exist on server
      theme string Optional //Theme name such as "standard", must exist on server
      timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
      mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
      description string Optional //User profile description
      descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      city string Optional //Home city of the user
      country string Optional //Home country code of the user, such as AU or CZ
      profileimageurlsmall string //User image profile URL - small version
      profileimageurl string //User image profile URL - big version
      customfields Optional //User custom fields (also known as user profile fields)
      list of (
        object {
          type string //The type of the custom field - text field, checkbox...
          value string //The value of the custom field (as stored in the database)
          displayvalue string Optional //The value of the custom field for display
          name string //The name of the custom field
          shortname string //The shortname of the custom field - to be able to build the field class in the code
        }
      )preferences Optional //Users preferences
      list of (
        object {
          name string //The name of the preferences
          value string //The value of the preference
        }
      )})
      )warnings Optional //List of warnings
      list of (
        //warning
        object {
          item string Optional //always set to 'key'
          itemid int Optional //faulty key name
          warningcode string //the warning code can be used by the client app to implement specific behaviour
          message string //untranslated english message to explain the warning
        }
      )})
}

```



XML-RPC (PHP structure)

```

Array
(
    [
        [users] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [username] => string
                                [firstname] => string
                                [lastname] => string
                                [fullname] => string
                                [email] => string
                                [address] => string
                                [phone1] => string
                                [phone2] => string
                                [department] => string
                                [institution] => string
                                [idnumber] => string
                                [interests] => string
                                [firstaccess] => int
                                [lastaccess] => int
                                [auth] => string
                                [suspended] => int
                                [confirmed] => int
                                [lang] => string
                                [calendartype] => string
                                [theme] => string
                                [timezone] => string
                                [mailformat] => int
                                [description] => string
                                [descriptionformat] => int
                                [city] => string
                                [country] => string
                                [profileimageurlsmall] => string
                                [profileimageurl] => string
                                [customfields] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [
                                                            [type] => string
                                                            [value] => string
                                                            [displayvalue] => string
                                                            [name] => string
                                                            [shortname] => string
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                [preferences] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [
                                                            [name] => string
                                                            [value] => string
                                                        )
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                        )
                )
            )
        )
    )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="username">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="firstname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="lastname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="email">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="address">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="phone1">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="phone2">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="department">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="institution">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="interests">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="firstaccess">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lastaccess">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="auth">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="suspended">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="confirmed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lang">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="calendartype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="theme">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timezone">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="mailformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="city">
    <VALUE>string</VALUE>
</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_get_users_by_field

Retrieve users' information for a specified unique field - If you want to do a user search, use core_user_get_users() or core_user_search_identity().

Arguments**field** (Required)

the search field can be 'id' or 'idnumber' or 'username' or 'email'

General structure

```
string //the search field can be
      'id' or 'idnumber' or 'username' or 'email'
```

XML-RPC (PHP structure)

```
[field] => string
```

REST (POST parameters)

```
field= string
```

values (Required)**General structure**

```
list of (
string //the value to match
)
```

XML-RPC (PHP structure)

```
[values] =>
  Array
  (
    [0] => string
  )
```

REST (POST parameters)

```
values[0]= string
```

Response

General structure

```

list of (
object {
id int //ID of the user
username string Optional //The username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //An email address - allow email as root@localhost
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //An arbitrary ID code number perhaps from the institution
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
auth string Optional //Auth plugins include manual, Ldap, etc
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
lang string Optional //Language code such as "en", must exist on server
calendartype string Optional //Calendar type such as "gregorian", must exist on server
theme string Optional //Theme name such as "standard", must exist on server
timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
description string Optional //User profile description
descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string //User image profile URL - small version
profileimageurl string //User image profile URL - big version
customfields Optional //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field (as stored in the database)
displayvalue string Optional //The value of the custom field for display
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
}
)preferences Optional //Users preferences
list of (
object {
name string //The name of the preferences
value string //The value of the preference
}
)
)
)

```



XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [suspended] => int
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [type] => string
                                        [value] => string
                                        [displayvalue] => string
                                        [name] => string
                                        [shortname] => string
                                    )
                        )
                [preferences] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [name] => string
                                        [value] => string
                                    )
                        )
                )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="auth">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="suspended">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="confirmed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="calendartype">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="theme">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timezone">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="mailformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
```



```

</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_user_remove_user_device

Remove a user device from the Moodle database.

Arguments**uuid** (Required)

the device UUID

General structure

```
string //the device UUID
```

XML-RPC (PHP structure)

```
[uuid] => string
```

REST (POST parameters)

```
uuid= string
```

appid (Default to "")

the app id, if empty devices matching the UUID for the user will be removed

General structure

```
string Default to "" //the app id, if empty devices matching the UUID for the user will be removed
```

XML-RPC (PHP structure)

```
[appid] => string
```

REST (POST parameters)

```
appid= string
```

Response**General structure**

```
object {
    removed int //True if removed, false if not removed because it doesn't exists
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
  (
    [removed] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
}
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="removed">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_search_identity

Return list of users identities matching the given criteria in their name or other identity fields.

Arguments

query (Required)

The search query

General structure

```
string //The search query
```

XML-RPC (PHP structure)

```
[query] => string
```

REST (POST parameters)

```
query= string
```

Response

General structure

```
object {
list list of (
object {
id int //ID of the user
fullname string //The fullname of the user
extrafields Optional //List of extra fields
list of (
object {
name string //Name of the extrafield.
value string //Value of the extrafield.
}
)
)
maxusersperpage int //Configured maximum users per page.
overflow int //Were there more records than maxusersperpage found?
}
```

XML-RPC (PHP structure)

```
Array
(
    [list] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [fullname] => string
                            [extrafields] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [name] => string
                                                    [value] => string
                                                )
                                    )
                )
            )
        [maxusersperpage] => int
        [overflow] => int
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="list">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="extrafields">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="value">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="maxusersperpage">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="overflow">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_user_set_user_preferences

Set user preferences.

Arguments

preferences (Required)

General structure

```
list of (
object {
name string //The name of the preference
value string //The value of the preference
userid int //Id of the user to set the preference
}
)
```

XML-RPC (PHP structure)

```
[preferences] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
[userid] => int
)
)
```

REST (POST parameters)

```
preferences[0][name]= string
preferences[0][value]= string
preferences[0][userid]= int
```

Response**General structure**

```
object {
saved //Preferences saved
list of (
object {
name string //The name of the preference
userid int //The user the preference was set for
}
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [saved] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [name] => string
                                [userid] => int
                            )
                        )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                        )
                )
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="saved">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_user_update_picture

Update or delete the user picture in the site

Arguments**draftitemid** (Required)

Id of the user draft file to use as image

General structure

```
int //Id of the user draft file to use as image
```

XML-RPC (PHP structure)

```
[draftitemid] => int
```

REST (POST parameters)

```
draftitemid= int
```

delete (Default to "")

If we should delete the user picture

General structure

```
int Default to "" //If we should delete the user picture
```

**XML-RPC (PHP structure)**

```
[delete] => int
```

REST (POST parameters)

```
delete= int
```

userid (Default to "0")

Id of the user, 0 for current user

General structure

```
int Default to "0" //Id of the user, 0 for current user
```

XML-RPC (PHP structure)

[userid] => int

REST (POST parameters)

userid= int

Response**General structure**

```
object {
    success int //True if the image was updated, false otherwise.
    profileimageurl string Optional //New profile user image url
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [success] => int
    [profileimageurl] => string
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="success">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="profileimageurl">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_update_user_device_public_key

Store mobile user public key.

Arguments**uuid** (Required)

the device UUID

General structure

```
string //the device UUID
```

XML-RPC (PHP structure)

```
[uuid] => string
```

REST (POST parameters)

```
uuid= string
```

appid (Required)

The app id, something like com.moodle.moodlemobile

General structure

```
string //The app id, something like com.moodle.moodlemobile
```

XML-RPC (PHP structure)

```
[appid] => string
```

REST (POST parameters)

```
appid= string
```

publickey (Required)

the app generated public key

General structure

```
string //the app generated public key
```

XML-RPC (PHP structure)

```
[publickey] => string
```

REST (POST parameters)

```
publickey= string
```

Response**General structure**

```
object {
status int //Whether the request was successful
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```

XML-RPC (PHP structure)

```
Array
  (
    [status] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_update_user_preferences

Update a user's preferences

Arguments

userid (Default to "0")

id of the user, default to current user

General structure

```
int Default to "0" //id of the user, default to current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

emailstop (Default to "null")

Enable or disable notifications for this user

General structure

```
int Default to "null" //Enable or disable notifications for this user
```

XML-RPC (PHP structure)

```
[emailstop] => int
```

REST (POST parameters)

```
emailstop= int
```

preferences (Default to "Array ()")

User preferences

General structure

```
Default to "Array"
(
)
" //User preferences
list of (
object {
type string //The name of the preference
value string Default to "null" //The value of the preference, do not set this field if you
want to remove (unset) the current value.
}
)
```



XML-RPC (PHP structure)

```
[preferences] =>
Array
(
[0] =>
Array
(
[type] => string
[value] => string
)
)
```

REST (POST parameters)

```
preferences[0][type]= string  
preferences[0][value]= string
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>  
<EXCEPTION class="invalid_parameter_exception">  
  <MESSAGE>Invalid parameter value detected</MESSAGE>  
  <DEBUGINFO></DEBUGINFO>  
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_user_update_users

Update users.

Arguments**users** (Required)

```
General structure

list of (
object {
id int //ID of the user
username string Optional //Username policy is defined in Moodle security config.
auth string Optional //Auth plugins include manual, Ldap, etc
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
password string Optional //Plain text password consisting of any characters
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
email string Optional //A valid and unique email address
maildisplay int Optional //Email visibility
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
description string Optional //User profile description, no HTML
userpicture int Optional //The itemid where the new user picture has been uploaded to, 0 to delete
firstnamephonetic string Optional //The first name(s) phonetically of the user
lastnamephonetic string Optional //The family name phonetically of the user
middlename string Optional //The middle name of the user
alternatename string Optional //The alternate name of the user
interests string Optional //User interests (separated by commas)
idnumber string Optional //An arbitrary ID code number perhaps from the institution
institution string Optional //Institution
department string Optional //Department
phone1 string Optional //Phone
phone2 string Optional //Mobile phone
address string Optional //Postal address
lang string Optional //Language code such as "en", must exist on server
calendartype string Optional //Calendar type such as "gregorian", must exist on server
theme string Optional //Theme name such as "standard", must exist on server
mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
customfields Optional //User custom fields (also known as user profil fields)
list of (
object {
type string //The name of the custom field
value string //The value of the custom field
})
preferences Optional //User preferences
list of (
object {
type string //The name of the preference
value string //The value of the preference
})
})
```



XML-RPC (PHP structure)

```
[users] =>
  Array
    (
      [0] =>
        Array
          (
            [
              [id] => int
              [username] => string
              [auth] => string
              [suspended] => int
              [password] => string
              [firstname] => string
              [lastname] => string
              [email] => string
              [maildisplay] => int
              [city] => string
              [country] => string
              [timezone] => string
              [description] => string
              [userpicture] => int
              [firstnamephonetic] => string
              [lastnamephonetic] => string
              [middlename] => string
              [alternatename] => string
              [interests] => string
              [idnumber] => string
              [institution] => string
              [department] => string
              [phone1] => string
              [phone2] => string
              [address] => string
              [lang] => string
              [calendartype] => string
              [theme] => string
              [mailformat] => int
              [customfields] =>
                Array
                  (
                    [0] =>
                      Array
                        (
                          [
                            [type] => string
                            [value] => string
                          )
                        )
                  )
                )
              [preferences] =>
                Array
                  (
                    [0] =>
                      Array
                        (
                          [
                            [type] => string
                            [value] => string
                          )
                        )
                  )
                )
              )
            )
          )
        )
      )
    )
  )
)
```

REST (POST parameters)

```

users[0][id]= int
users[0][username]= string
users[0][auth]= string
users[0][suspended]= int
users[0][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][maildisplay]= int
users[0][city]= string
users[0][country]= string
users[0][timezone]= string
users[0][description]= string
users[0][userpicture]= int
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
users[0][interests]= string
users[0][idnumber]= string
users[0][institution]= string
users[0][department]= string
users[0][phone1]= string
users[0][phone2]= string
users[0][address]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][mailformat]= int
users[0][customfields][0][type]= string
users[0][customfields][0][value]= string
users[0][preferences][0][type]= string
users[0][preferences][0][value]= string

```

Response**General structure**

```

object {
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}
```

XML-RPC (PHP structure)

```

Array
(
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_user_view_user_list

Simulates the web-interface view of user/index.php (triggering events)..

Arguments**courseid** (Required)

id of the course, 0 for site

General structure

```
int //id of the course, 0 for site
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_user_view_user_profile

Simulates the web-interface view of user/view.php and user/profile.php (triggering events),.

Arguments**userid** (Required)

id of the user, 0 for current user

General structure

```
int //id of the user, 0 for current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

courseid (Default to "0")

id of the course, default site course

General structure

```
int Default to "0" //id of the course, default site course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```

object {
    status int   //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string  Optional //item
        itemid int  Optional //item id
        warningcode string  //the warning code can be used by the client app to implement specific behaviour
        message string  //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_webservice_get_site_info

Return some site info / user info / list web service functions

Arguments

serviceshortnames (Default to "Array ()")

DEPRECATED PARAMETER - it was a design error in the original implementation. \ It is ignored now. (parameter kept for backward compatibility)

```
General structure
Default to "Array
(
)
" //DEPRECATED PARAMETER - it was a design error in the original implementation. \
It is ignored now. (parameter kept for backward compatibility)
list of (
string //service shortname
)
```

XML-RPC (PHP structure)

```
[serviceshortnames] =>
Array
(
    [0] => string
)
```

REST (POST parameters)

```
serviceshortnames[0]= string
```

Response



General structure

```

object {
    sitename string //site name
    username string //username
    firstname string //first name
    lastname string //last name
    fullname string //user full name
    lang string //Current Language.
    userid int //user id
    siteurl string //site url
    userpictureurl string //the user profile picture.

        Warning: this url is the public URL that only works when forcedlogin is set to NO and guestaccess is set to YES.
        In order to retrieve user profile pictures independently of the Moodle config, replace "pluginfile.php" by
        "webservice/pluginfile.php?token=WSTOKEN&file="
        Of course the user can only see profile picture depending
        on his/her permissions. Moreover it is recommended to use HTTPS too.

functions list of (
    //functions that are available
object {
    name string //function name
    version string //The version number of the component to which the function belongs
}
)downloadfiles int Optional //1 if users are allowed to download files, 0 if not
uploadfiles int Optional //1 if users are allowed to upload files, 0 if not
release string Optional //Moodle release number
version string Optional //Moodle version number
mobilecssurl string Optional //Mobile custom CSS theme
advancedfeatures Optional //Advanced features availability
list of (
    //Advanced features availability
object {
    name string //feature name
    value int //feature value. Usually 1 means enabled.
}
)usercanmanageownfiles int Optional //true if the user can manage his own files
userquota int Optional //user quota (bytes). 0 means user can ignore the quota
usermaxuploadfilesize int Optional //user max upload file size (bytes). -1 means the user can ignore the upload file size
userhomepage int Optional //the default home page for the user: 0 for the site home, 1 for dashboard
userprivateaccesskey string Optional //Private user access key for fetching files.
siteid int Optional //Site course ID
sitecalendartype string Optional //Calendar type set in the site.
usercalendartype string Optional //Calendar typed used by the user.
userissiteadmin int Optional //Whether the user is a site admin or not.
theme string Optional //Current theme for the user.
limitconcurrentlogins int Optional //Number of concurrent sessions allowed
usersessionscount int Optional //Number of active sessions for current user.
    Only returned when limitconcurrentlogins is used.
}

```



XML-RPC (PHP structure)

```
Array
(
    [sitename] => string
    [username] => string
    [firstname] => string
    [lastname] => string
    [fullname] => string
    [lang] => string
    [userid] => int
    [siteurl] => string
    [userpictureurl] => string
    [functions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [version] => string
                        )
            )
    [downloadfiles] => int
    [uploadfiles] => int
    [release] => string
    [version] => string
    [mobilecssurl] => string
    [advancedfeatures] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [value] => int
                        )
            )
    [usercanmanageownfiles] => int
    [userquota] => int
    [usermaxuploadfilesize] => int
    [userhomepage] => int
    [userprivateaccesskey] => string
    [siteid] => int
    [sitecalendartype] => string
    [usercalendartype] => string
    [userissiteadmin] => int
    [theme] => string
    [limitconcurrentlogins] => int
    [usersessionscount] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="sitename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="username">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="firstname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="lastname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="fullname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="lang">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="userid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="siteurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="userpictureurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="functions">
      <MULTIPLE>
        <SINGLE>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="version">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="downloadfiles">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="uploadfiles">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="release">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="version">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="mobilecssurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="advancedfeatures">
      <MULTIPLE>
        <SINGLE>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="value">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="usercanmanageownfiles">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userquota">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="usermaxuploadfilesize">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userhomepage">
      <VALUE>int</VALUE>
    </KEY>
```



```

<KEY name="userprivateaccesskey">
    <VALUE>string</VALUE>
</KEY>
<KEY name="siteid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="sitecalendartype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usercalendartype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="userissiteadmin">
    <VALUE>int</VALUE>
</KEY>
<KEY name="theme">
    <VALUE>string</VALUE>
</KEY>
<KEY name="limitconcurrentlogins">
    <VALUE>int</VALUE>
</KEY>
<KEY name="usersessionscount">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

core_xapi_delete_state



Delete an xAPI state data from an activityId.

Arguments

component (Required)

Component name

General structure

```
string //Component name
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

activityId (Required)

xAPI activity ID IRI

General structure

```
string //xAPI activity ID IRI
```

XML-RPC (PHP structure)

```
[activityId] => string
```

REST (POST parameters)

```
activityId= string
```

agent (Required)

The xAPI agent json

General structure

```
string //The xAPI agent json
```

XML-RPC (PHP structure)

```
[agent] => string
```

REST (POST parameters)

```
agent= string
```

stateId (Required)

The xAPI state ID

General structure

```
string //The xAPI state ID
```

**XML-RPC (PHP structure)**

```
[stateId] => string
```

REST (POST parameters)

```
stateId= string
```

registration (Default to "null")

The xAPI registration UUID

General structure

```
string Default to "null" //The xAPI registration UUID
```

XML-RPC (PHP structure)

```
[registration] => string
```

REST (POST parameters)

```
registration= string
```

Response

If the state data is deleted

General structure

```
int //If the state data is deleted
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes



core_xapi_get_state

Get an xAPI state data from an activityId.

Arguments

component (Required)

Component name

General structure

```
string //Component name
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

activityId (Required)

xAPI activity ID IRI

General structure

```
string //xAPI activity ID IRI
```

XML-RPC (PHP structure)

```
[activityId] => string
```

REST (POST parameters)

```
activityId= string
```

agent (Required)

The xAPI agent json

General structure

```
string //The xAPI agent json
```

XML-RPC (PHP structure)

```
[agent] => string
```

REST (POST parameters)

```
agent= string
```

stateId (Required)

The xAPI state ID

General structure

```
string //The xAPI state ID
```

**XML-RPC (PHP structure)**

```
[stateId] => string
```

REST (POST parameters)

```
stateId= string
```

registration (Default to "null")

The xAPI registration UUID

General structure

```
string Default to "null" //The xAPI registration UUID
```

XML-RPC (PHP structure)

```
[registration] => string
```

REST (POST parameters)

```
registration= string
```

Response

The state data json

General structure

```
string //The state data json
```

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes



core_xapi_get_states

Get all state ID from an activityId.

Arguments

component (Required)

Component name

General structure

```
string //Component name
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

activityId (Required)

xAPI activity ID IRI

General structure

```
string //xAPI activity ID IRI
```

XML-RPC (PHP structure)

```
[activityId] => string
```

REST (POST parameters)

```
activityId= string
```

agent (Required)

The xAPI agent json

General structure

```
string //The xAPI agent json
```

XML-RPC (PHP structure)

```
[agent] => string
```

REST (POST parameters)

```
agent= string
```

registration (Default to "null")

The xAPI registration UUID

General structure

```
string Default to "null" //The xAPI registration UUID
```

**XML-RPC (PHP structure)**

```
[registration] => string
```

REST (POST parameters)

```
registration= string
```

since (Default to "null")

Filter ids stored since the timestamp (exclusive)

General structure

```
string Default to "null" //Filter ids stored since the timestamp (exclusive)
```

XML-RPC (PHP structure)

```
[since] => string
```

REST (POST parameters)

```
since= string
```

Response

List of state Ids

General structure

```
//List of state IDs
list of (
string //State ID
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <VALUE>string</VALUE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

Yes

core_xapi_post_state

Post an xAPI state into an activityId.

Arguments

component (Required)

Component name

General structure

```
string //Component name
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

activityId (Required)

xAPI activity ID IRI

General structure

```
string //xAPI activity ID IRI
```

XML-RPC (PHP structure)

```
[activityId] => string
```

REST (POST parameters)

```
activityId= string
```

agent (Required)

The xAPI agent json

General structure

```
string //The xAPI agent json
```

XML-RPC (PHP structure)

```
[agent] => string
```

REST (POST parameters)

```
agent= string
```



stateId (Required)

The xAPI state ID

General structure

```
string //The xAPI state ID
```

XML-RPC (PHP structure)

```
[stateId] => string
```

REST (POST parameters)

```
stateId= string
```

stateData (Required)

JSON object with the state data

General structure

```
string //JSON object with the state data
```

XML-RPC (PHP structure)

```
[stateData] => string
```

REST (POST parameters)

```
stateData= string
```

registration (Default to "null")

The xAPI registration UUID

General structure

```
string Default to "null" //The xAPI registration UUID
```

XML-RPC (PHP structure)

```
[registration] => string
```

REST (POST parameters)

```
registration= string
```

Response

If the state is accepted

General structure

```
int //If the state is accepted
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

core_xapi_statement_post

Post an xAPI statement.

Arguments**component** (Required)

Component name

General structure`string //Component name`**XML-RPC (PHP structure)**`[component] => string`**REST (POST parameters)**`component= string`**requestjson** (Required)

json object with all the statements to post

General structure`string //json object with all the statements to post`**XML-RPC (PHP structure)**`[requestjson] => string`**REST (POST parameters)**`requestjson= string`**Response**

List of statements storing acceptance results

General structure

```
//List of statements storing acceptance results
list of (
int //If the statement is accepted
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <VALUE>int</VALUE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

enrol_guest_get_instance_info

Return guest enrolment instance information.

Arguments**instanceid (Required)**

Instance id of guest enrolment plugin.

General structure

```
int //Instance id of guest enrolment plugin.
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

Response

General structure

```
object {
    instanceinfo object {
        id int   //Id of course enrolment instance
        courseid int //Id of course
        type string //Type of enrolment plugin
        name string //Name of enrolment plugin
        status int //Is the enrolment enabled?
        passwordrequired int //Is a password required?
    }
    warnings Optional //List of warnings
    list of (
        //warning
    )
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```

XML-RPC (PHP structure)



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="instanceinfo">
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="courseid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="type">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="status">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="passwordrequired">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

enrol_manual_enrol_users

Manual enrol users

Arguments

enrolments (Required)

General structure

```
list of (
object {
roleid int //Role to assign to the user
userid int //The user that is going to be enrolled
courseid int //The course to enrol the user role in
timestart int Optional //Timestamp when the enrolment start
timeend int Optional //Timestamp when the enrolment end
suspend int Optional //set to 1 to suspend the enrolment
}
)
```

XML-RPC (PHP structure)

```
[enrolments] =>
  Array
    (
      [0] =>
        Array
          (
            [roleid] => int
            [userid] => int
            [courseid] => int
            [timestart] => int
            [timeend] => int
            [suspend] => int
          )
    )
)
```

REST (POST parameters)

```
enrolments[0][roleid]= int
enrolments[0][userid]= int
enrolments[0][courseid]= int
enrolments[0][timestart]= int
enrolments[0][timeend]= int
enrolments[0][suspend]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

enrol_manual_unenrol_users

Manual unenrol users

Arguments

enrolments (Required)

General structure

```
list of (
object {
userid int //The user that is going to be unenrolled
courseid int //The course to unenrol the user from
roleid int Optional //The user role
}
)
```

XML-RPC (PHP structure)

```
[enrolments] =>
  Array
    (
      [0] =>
        Array
          (
            [userid] => int
            [courseid] => int
            [roleid] => int
          )
    )
)
```

REST (POST parameters)

```
enrolments[0][userid]= int
enrolments[0][courseid]= int
enrolments[0][roleid]= int
```

Response**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

enrol_meta_add_instances

Add meta enrolment instances

Arguments

instances (Default to "Array ()")

List of course meta enrolment instances to create.

General structure

```
Default to "Array
(
)
" //List of course meta enrolment instances to create.
list of (
object {
metacourseid int  //ID of the course where meta enrolment is added.
courseid string  //ID of the course where meta enrolment is linked to.
creategroup int Default to "" //Creates group in meta course named after linked course and puts all enrolled users in this group
}
)
```

XML-RPC (PHP structure)

```
[instances] =>
Array
(
[0] =>
Array
(
[metacourseid] => int
[courseid] => string
[creategroup] => int
)
)
```

REST (POST parameters)

```
instances[0][metacourseid]= int
instances[0][courseid]= string
instances[0][creategroup]= int
```

Response

List of course meta enrolment instances that were created.

General structure

```
Default to "Array
(
)
" //List of course meta enrolment instances that were created.
list of (
object {
metacourseid int  //ID of the course where meta enrolment is added.
courseid string  //ID of the course where meta enrolment is linked to.
status int  //True on success, false if link already exists.
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
Array
(
[metacourseid] => int
[courseid] => string
[status] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="metacourseid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="courseid">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="status">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

enrol_meta_delete_instances

Delete meta enrolment instances

Arguments**instances** (Default to "Array ()")

List of course meta enrolment instances to delete.

General structure

```
Default to "Array
(
)
" //List of course meta enrolment instances to delete.
list of (
object {
  metacourseid int  //ID of the course with meta enrolment.
  courseid string   //ID of the course where meta enrolment is Linked to.
})
```

XML-RPC (PHP structure)

```
[instances] =>
  Array
  (
    [0] =>
      Array
      (
        [metacourseid] => int
        [courseid] => string
      )
  )
)
```

REST (POST parameters)

```
instances[0][metacourseid]= int
instances[0][courseid]= string
```

Response

List of course meta enrolment instances that were deleted.

General structure

```
Default to "Array"
(
)
" //List of course meta enrolment instances that were deleted.
list of (
object {
metacourseid int  //ID of the course where meta enrolment is deleted.
courseid string  //ID of the course that was meta linked.
status int  //True on success, false if meta Link did not exist.
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
  Array
  (
    [metacourseid] => int
    [courseid] => string
    [status] => int
  )
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="metacourseid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="courseid">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="status">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

enrol_self_enrol_user

Self enrol the current user in the given course.

Arguments**courseid** (Required)

Id of the course

General structure

```
int //Id of the course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

password (Default to "")

Enrolment key

General structure

```
string Default to "" //Enrolment key
```

XML-RPC (PHP structure)

```
[password] => string
```

REST (POST parameters)

```
password= string
```

instanceid (Default to "0")

Instance id of self enrolment plugin.

General structure

```
int Default to "0" //Instance id of self enrolment plugin.
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```

Response**General structure**

```
object {
    status int //status: true if the user is enrolled, false otherwise
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

enrol_self_get_instance_info

self enrolment instance information.

Arguments

instanceid (Required)

instance id of self enrolment plugin.

General structure

```
int //instance id of self enrolment plugin.
```

XML-RPC (PHP structure)

```
[instanceid] => int
```

REST (POST parameters)

```
instanceid= int
```



Response

General structure

```
object {
id int //id of course enrolment instance
courseid int //id of course
type string //type of enrolment plugin
name string //name of enrolment plugin
status string //status of enrolment plugin
enrolpassword string Optional //password required for enrolment
}
```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [courseid] => int
    [type] => string
    [name] => string
    [status] => string
    [enrolpassword] => string
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="type">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="name">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="status">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="enrolpassword">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_grader_get_users_in_report

Returns the dataset of users within the report

Arguments

courseid (Required)

Course ID

General structure

```
int //Course ID
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```

object {
  users list of (
    object {
      id int //ID of the user
      username string Optional //The username
      firstname string Optional //The first name(s) of the user
      lastname string Optional //The family name of the user
      fullname string //The fullname of the user
      email string Optional //An email address - allow email as root@localhost
      address string Optional //Postal address
      phone1 string Optional //Phone 1
      phone2 string Optional //Phone 2
      department string Optional //department
      institution string Optional //institution
      idnumber string Optional //An arbitrary ID code number perhaps from the institution
      interests string Optional //user interests (separated by commas)
      firstaccess int Optional //first access to the site (0 if never)
      lastaccess int Optional //last access to the site (0 if never)
      auth string Optional //Auth plugins include manual, Ldap, etc
      suspended int Optional //Suspend user account, either false to enable user login or true to disable it
      confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
      lang string Optional //Language code such as "en", must exist on server
      calendartype string Optional //Calendar type such as "gregorian", must exist on server
      theme string Optional //Theme name such as "standard", must exist on server
      timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
      mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
      description string Optional //User profile description
      descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      city string Optional //Home city of the user
      country string Optional //Home country code of the user, such as AU or CZ
      profileimageurlsmall string //User image profile URL - small version
      profileimageurl string //User image profile URL - big version
      customfields Optional //User custom fields (also known as user profile fields)
      list of (
        object {
          type string //The type of the custom field - text field, checkbox...
          value string //The value of the custom field (as stored in the database)
          displayvalue string Optional //The value of the custom field for display
          name string //The name of the custom field
          shortname string //The shortname of the custom field - to be able to build the field class in the code
        }
      )preferences Optional //Users preferences
      list of (
        object {
          name string //The name of the preferences
          value string //The value of the preference
        }
      )})
      )warnings Optional //List of warnings
      list of (
        //warning
        object {
          item string Optional //item
          itemid int Optional //item id
          warningcode string //the warning code can be used by the client app to implement specific behaviour
          message string //untranslated english message to explain the warning
        }
      )})
}

```



XML-RPC (PHP structure)

```

Array
(
    [
        [users] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [username] => string
                                [firstname] => string
                                [lastname] => string
                                [fullname] => string
                                [email] => string
                                [address] => string
                                [phone1] => string
                                [phone2] => string
                                [department] => string
                                [institution] => string
                                [idnumber] => string
                                [interests] => string
                                [firstaccess] => int
                                [lastaccess] => int
                                [auth] => string
                                [suspended] => int
                                [confirmed] => int
                                [lang] => string
                                [calendartype] => string
                                [theme] => string
                                [timezone] => string
                                [mailformat] => int
                                [description] => string
                                [descriptionformat] => int
                                [city] => string
                                [country] => string
                                [profileimageurlsmall] => string
                                [profileimageurl] => string
                                [customfields] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [
                                                            [type] => string
                                                            [value] => string
                                                            [displayvalue] => string
                                                            [name] => string
                                                            [shortname] => string
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                [preferences] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [
                                                            [name] => string
                                                            [value] => string
                                                        )
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                        )
                )
            )
        )
    )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="username">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="firstname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="lastname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="email">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="address">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="phone1">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="phone2">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="department">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="institution">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="idnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="interests">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="firstaccess">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lastaccess">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="auth">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="suspended">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="confirmed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="lang">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="calendartype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="theme">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timezone">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="mailformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="descriptionformat">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="city">
    <VALUE>string</VALUE>
</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

gradereport_overview_get_course_grades

Get the given user courses final grades

Arguments**userid** (Default to "0")

Get grades for this user (optional, default current)

General structure

```
int Default to "0" //Get grades for this user (optional, default current)
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
grades list of (
object {
courseid int //Course id
grade string //Grade formatted
rawgrade string //Raw grade, not formatted
rank int Optional //Your rank in the course
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [grades] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [courseid] => int
                                [grade] => string
                                [rawgrade] => string
                                [rank] => int
                            )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="grades">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="grade">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rawgrade">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="rank">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_overview_view_grade_report

Trigger the report view event

Arguments**courseid** (Required)

id of the course

General structure

```
int //id of the course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

id of the user, 0 means current user

General structure

```
int Default to "0" //id of the user, 0 means current user
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response

General structure

```

object {
    status int    //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string  Optional //item
        itemid int   Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_singleview_get_grade_items_for_search_widget

Get the gradeitem(s) for a course

Arguments

courseid (Required)

Course Id

General structure

```
int //Course Id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
object {
    gradeitems list of (
        object {
            id int Optional //ID of the grade item
            name string Optional //The full name of the grade item
        }
    )warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```



XML-RPC (PHP structure)

```

Array
(
    [
        [gradeitems] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [
                            [id] => int
                            [name] => string
                        )
                    )
            )
        [warnings] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="gradeitems">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

gradereport_user_get_access_information

Returns user access information for the user grade report.

Arguments**courseid** (Required)

Course to check.

General structure

```
int //Course to check.
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
    canviewusergradereport int //Whether the user can view the user grade report.
    canviewmygrades int //Whether the user can view his grades in the course.
    canviewallgrades int //Whether the user can view all users grades in the course.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```

Array
(
    [canviewusergradereport] => int
    [canviewmygrades] => int
    [canviewallgrades] => int
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
        )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="canviewusergradereport">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewmygrades">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewallgrades">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_user_get_grade_items

Returns the complete list of grade items for users in a course

Arguments

courseid (Required)

Course Id

General structure

```
int //Course Id
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

Return grades only for this user (optional)

General structure

```
int Default to "0" //Return grades only for this user (optional)
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

groupid (Default to "0")

Get users from this group only



General structure

```
int Default to "0" //Get users from this group only
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response

General structure

```

object {
  usergrades list of (
    object {
      courseid int //course id
      courseidnumber string //course idnumber
      userid int //user id
      userfullname string //user fullname
      useridnumber string //user idnumber
      maxdepth int //table max depth (needed for printing it)
      gradeitems list of (
        //Grade items
      object {
        id int //Grade item id
        itemname string //Grade item name
        itemtype string //Grade item type
        itemmodule string //Grade item module
        iteminstance int //Grade item instance
        itemnumber int //Grade item item number
        idnumber string //Grade item idnumber
        categoryid int //Grade item category id
        outcomeid int //Outcome id
        scaleid int //Scale id
        locked int Optional //Grade item for user locked?
        cmid int Optional //Course module id (if type mod)
        weightraw double Optional //Weight raw
        weightformatted string Optional //Weight
        status string Optional //Status
        graderaw double Optional //Grade raw
        gradedatesubmitted int Optional //Grade submit date
        gradedategraded int Optional //Grade graded date
        gradehiddenbydate int Optional //Grade hidden by date?
        gradeneedsupdate int Optional //Grade needs update?
        gradeishidden int Optional //Grade is hidden?
        gradeislocked int Optional //Grade is locked?
        gradeisoverridden int Optional //Grade overridden?
        gradeformatted string Optional //The grade formatted
        grademin double Optional //Grade min
        grademax double Optional //Grade max
        rangeformatted string Optional //Range formatted
        percentageformatted string Optional //Percentage
        lettergradeformatted string Optional //Letter grade
        rank int Optional //Rank in the course
        numusers int Optional //Num users in course
        averageformatted string Optional //Grade average
        feedback string Optional //Grade feedback
        feedbackformat int Optional //feedback format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      }
    })
  )
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}

```



XML-RPC (PHP structure)

```

Array
(
    [usergrades] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [courseid] => int
                            [courseidnumber] => string
                            [userid] => int
                            [userfullname] => string
                            [useridnumber] => string
                            [maxdepth] => int
                            [gradeitems] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [itemname] => string
                                                    [itemtype] => string
                                                    [itemmodule] => string
                                                    [iteminstance] => int
                                                    [itemnumber] => int
                                                    [idnumber] => string
                                                    [categoryid] => int
                                                    [outcomeid] => int
                                                    [scaleid] => int
                                                    [locked] => int
                                                    [cmid] => int
                                                    [weightraw] => double
                                                    [weightformatted] => string
                                                    [status] => string
                                                    [graderaw] => double
                                                    [gradedatesubmitted] => int
                                                    [gradedategraded] => int
                                                    [gradehiddenbydate] => int
                                                    [gradeneedsupdate] => int
                                                    [gradeishidden] => int
                                                    [gradeislocked] => int
                                                    [gradeisoverridden] => int
                                                    [gradeformatted] => string
                                                    [grademin] => double
                                                    [grademax] => double
                                                    [rangeformatted] => string
                                                    [percentageformatted] => string
                                                    [lettergradeformatted] => string
                                                    [rank] => int
                                                    [numusers] => int
                                                    [averageformatted] => string
                                                    [feedback] => string
                                                    [feedbackformat] => int
                                                )
                                        )
                                    )
                                )
                            )
                        )
                )
            )
)
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
            )
        )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="usergrades">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseidnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userfullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="useridnumber">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="maxdepth">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="gradeitems">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="itemname">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="itemtype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="itemmodule">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="iteminstance">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="itemnumber">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="idnumber">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="categoryid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="outcomeid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="scaleid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="locked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="cmid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="weightraw">
                                    <VALUE>double</VALUE>
                                </KEY>
                                <KEY name="weightformatted">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="status">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="graderaw">
                                    <VALUE>double</VALUE>
                                </KEY>
                                <KEY name="gradedatesubmitted">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="gradedategraded">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```

<KEY name="gradehiddenbydate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradeneedsupdate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradeishidden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradeislocked">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradeisoverridden">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="grademin">
    <VALUE>double</VALUE>
</KEY>
<KEY name="grademax">
    <VALUE>double</VALUE>
</KEY>
<KEY name="rangeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="percentageformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="lettergradeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="rank">
    <VALUE>int</VALUE>
</KEY>
<KEY name="numusers">
    <VALUE>int</VALUE>
</KEY>
<KEY name="averageformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="feedback">
    <VALUE>string</VALUE>
</KEY>
<KEY name="feedbackformat">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>

```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_user_get_grades_table

Get the user/s report grades table for a course

Arguments**courseid** (Required)

Course Id

General structure

int //Course Id

XML-RPC (PHP structure)

[courseid] => int

REST (POST parameters)

courseid= int

userid (Default to "0")

Return grades only for this user (optional)

General structure

int Default to "0" //Return grades only for this user (optional)

**XML-RPC (PHP structure)**

[userid] => int

REST (POST parameters)

userid= int

groupid (Default to "0")

Get users from this group only

General structure

int Default to "0" //Get users from this group only

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response



```
General structure

object {
  tables list of (
    object {
      courseid int    //course id
      userid int     //user id
      userfullname string //user fullname
      maxdepth int   //table max depth (needed for printing it)
      tabledata list of (
        //table
        object {
          itemname Optional //The item returned data
          object {
            class string //class
            colspan int //col span
            content string //cell content
            id string //id
          }
          leader Optional //The item returned data
          object {
            class string //class
            rowspan int //row span
          }
          weight Optional //weight column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          grade Optional //grade column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          range Optional //range column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          percentage Optional //percentage column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          lettergrade Optional //lettergrade column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          rank Optional //rank column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          average Optional //average column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          feedback Optional //feedback column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          contributiontocourse total Optional //contributiontocourse total column
          object {
            class string //class
            content string //cell content
            headers string //headers
          }
          parentcategories list of (
            int //Parent grade category ID.
          )}
        )}
      )})
    )})
  )})
}warnings Optional //list of warnings
```

```
list of (
  //warning
object {
item string  Optional //item
itemid int   Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)}
```



XML-RPC (PHP structure)

```

Array
(
    [tables] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [courseid] => int
                            [userid] => int
                            [userfullname] => string
                            [maxdepth] => int
                            [tabledata] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [itemname] =>
                                                        Array
                                                            (
                                                                [class] => string
                                                                [colspan] => int
                                                                [content] => string
                                                                [id] => string
                                                            )
                                            )
                                        [leader] =>
                                            Array
                                                (
                                                    [class] => string
                                                    [rowspan] => int
                                                )
                                    )
                            [weight] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [grade] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [range] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [percentage] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [lettergrade] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [rank] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                            [average] =>
                                Array
                                    (
                                        [class] => string
                                        [content] => string
                                        [headers] => string
                                    )
                        )
                )
            )
)

```





REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="tables">
      <MULTIPLE>
        <SINGLE>
          <KEY name="courseid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="userid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="userfullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="maxdepth">
            <VALUE>int</VALUE>
          </KEY>
        <KEY name="tableddata">
          <MULTIPLE>
            <SINGLE>
              <KEY name="itemname">
                <SINGLE>
                  <KEY name="class">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="colspan">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="content">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="id">
                    <VALUE>string</VALUE>
                  </KEY>
                </SINGLE>
              </KEY>
            <KEY name="leader">
              <SINGLE>
                <KEY name="class">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="rowspan">
                  <VALUE>int</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
            <KEY name="weight">
              <SINGLE>
                <KEY name="class">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="headers">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
            <KEY name="grade">
              <SINGLE>
                <KEY name="class">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="headers">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
            <KEY name="range">
              <SINGLE>
                <KEY name="class">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
          </MULTIPLE>
        </KEY>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```
<KEY name="headers">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="percentage">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="lettergrade">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="rank">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="average">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="feedback">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="contributiontocoursestotal">
    <SINGLE>
        <KEY name="class">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="content">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="headers">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```

<KEY name="parentcategories">
    <MULTIPLE>
        <VALUE>int</VALUE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

gradereport_user_view_grade_report



Trigger the report view event

Arguments

courseid (Required)

id of the course

General structure

```
int //id of the course
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

userid (Default to "0")

id of the user, 0 means current user

General structure

```
int Default to "0" //id of the user, 0 means current user
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    ]
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**gradingform_guide_grader_gradingpanel_fetch**

Fetch the data required to display the grader grading panel, creating the grade item if required

Arguments**component** (Required)

The name of the component

General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

**XML-RPC (PHP structure)**

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

Response

General structure

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        instanceid int //The id of the current grading instance
        criterion //The criterion by which this item will be graded
        list of (
            object {
                id int //The id of the criterion
                name string //The name of the criterion
                maxscore double //The maximum score for this criterion
                description string //The description of the criterion
                descriptionmarkers string //The description of the criterion for markers
                score double Optional //The current score for user being assessed
                remark string Optional //Any remarks for this criterion for the user being assessed
            }
        )hascomments int //Whether there are any frequently-used comments
        comments //Frequently used comments
        list of (
            object {
                id int //Comment id
                sortorder int //The sortorder of this comment
                description string //The comment value
            }
        )usergrade string //Current user grade
        maxgrade string //Max possible grade
        gradedby string //The assumed grader of this grading instance
        timecreated int //The time that the grade was created
        timemodified int //The time that the grade was last updated
    }
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [templatename] => string
        [hasgrade] => int
        [grade] =>
            Array
            (
                [
                    [instanceid] => int
                    [criterion] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [id] => int
                                            [name] => string
                                            [maxscore] => double
                                            [description] => string
                                            [descriptionmarkers] => string
                                            [score] => double
                                            [remark] => string
                                        )
                                    )
                )
            )
        [hascomments] => int
        [comments] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [sortorder] => int
                                [description] => string
                            )
                )
            )
        [usergrade] => string
        [maxgrade] => string
        [gradedby] => string
        [timecreated] => int
        [timemodified] => int
    )
)
[warnings] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="templatename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="hasgrade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="grade">
      <SINGLE>
        <KEY name="instanceid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="criterion">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="name">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="maxscore">
                <VALUE>double</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="descriptionmarkers">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="score">
                <VALUE>double</VALUE>
              </KEY>
              <KEY name="remark">
                <VALUE>string</VALUE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
        <KEY name="hascomments">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="comments">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="sortorder">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
        <KEY name="usergrade">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="maxgrade">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="gradedby">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="timecreated">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

```

</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

gradingform_guide_grader_gradingpanel_store

Store the grading data for a user from the grader grading panel.

Arguments

component (Required)

The name of the component

**General structure**

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

notifyuser (Default to "")

Whetehr to notify the user or not

General structure

```
int Default to "" //Whetehr to notify the user or not
```

XML-RPC (PHP structure)

```
[notifyuser] => int
```

REST (POST parameters)

```
notifyuser= int
```

formdata (Required)

The serialised form data representing the grade

General structure

```
string //The serialised form data representing the grade
```

XML-RPC (PHP structure)

```
[formdata] => string
```

REST (POST parameters)

```
formdata= string
```

Response**General structure**

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        instanceid int //The id of the current grading instance
        criterion //The criterion by which this item will be graded
        list of (
            object {
                id int //The id of the criterion
                name string //The name of the criterion
                maxscore double //The maximum score for this criterion
                description string //The description of the criterion
                descriptionmarkers string //The description of the criterion for markers
                score double Optional //The current score for user being assessed
                remark string Optional //Any remarks for this criterion for the user being assessed
            }
        )hascomments int //Whether there are any frequently-used comments
        comments //Frequently used comments
        list of (
            object {
                id int //Comment id
                sortorder int //The sortorder of this comment
                description string //The comment value
            }
        )usergrade string //Current user grade
        maxgrade string //Max possible grade
        gradedby string //The assumed grader of this grading instance
        timecreated int //The time that the grade was created
        timemodified int //The time that the grade was last updated
    }
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```



XML-RPC (PHP structure)

```
Array
(
    [templatename] => string
    [hasgrade] => int
    [grade] =>
        Array
        (
            [instanceid] => int
            [criterion] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [id] => int
                            [name] => string
                            [maxscore] => double
                            [description] => string
                            [descriptionmarkers] => string
                            [score] => double
                            [remark] => string
                        )
                )
            [hascomments] => int
            [comments] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [id] => int
                            [sortorder] => int
                            [description] => string
                        )
                )
            [usergrade] => string
            [maxgrade] => string
            [gradedby] => string
            [timecreated] => int
            [timemodified] => int
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="templatename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="hasgrade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="grade">
      <SINGLE>
        <KEY name="instanceid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="criterion">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="name">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="maxscore">
                <VALUE>double</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="descriptionmarkers">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="score">
                <VALUE>double</VALUE>
              </KEY>
              <KEY name="remark">
                <VALUE>string</VALUE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
        <KEY name="hascomments">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="comments">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="sortorder">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
        <KEY name="usergrade">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="maxgrade">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="gradedby">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="timecreated">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```

</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

gradingform_rubric_grader_gradingpanel_fetch

Fetch the data required to display the grader grading panel, creating the grade item if required

Arguments

component (Required)

The name of the component



General structure

```
string //The name of the component
```

XML-RPC (PHP structure)

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

Response

General structure

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        instanceid int //The id of the current grading instance
        rubricmode string //The mode i.e. evaluate editable
        canedit int //Can the user edit this
        criteria list of (
            object {
                id int //ID of the Criteria
                description string //Description of the Criteria
                remark string Optional //Any remarks for this criterion for the user being assessed
                levels list of (
                    object {
                        id int //ID of Level
                        criterionid int //ID of the criterion this matches to
                        score string //What this level is worth
                        definition string //Definition of the level
                        checked int //Selected flag
                    }
                ))
                timecreated int //The time that the grade was created
                usergrade string //Current user grade
                maxgrade string //Max possible grade
                gradedby string //The assumed grader of this grading instance
                timemodified int //The time that the grade was last updated
            }
        )
        warnings Optional //list of warnings
        list of (
            //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    ))
}
```



XML-RPC (PHP structure)

```
Array
(
    [
        [templatename] => string
        [hasgrade] => int
        [grade] =>
            Array
            (
                [
                    [instanceid] => int
                    [rubricmode] => string
                    [canedit] => int
                    [criteria] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [id] => int
                                            [description] => string
                                            [remark] => string
                                            [levels] =>
                                                Array
                                                (
                                                    [
                                                        [0] =>
                                                            Array
                                                            (
                                                                [
                                                                    [id] => int
                                                                    [criterionid] => int
                                                                    [score] => string
                                                                    [definition] => string
                                                                    [checked] => int
                                                                )
                                                            )
                                                    )
                                                )
                                        )
                                    )
                )
            )
        [
            [timecreated] => int
            [usergrade] => string
            [maxgrade] => string
            [gradedby] => string
            [timemodified] => int
        )
    ]
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="templatename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="hasgrade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="grade">
      <SINGLE>
        <KEY name="instanceid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="rubricmode">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="canedit">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="criteria">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="remark">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="levels">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="id">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="criterionid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="score">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="definition">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="checked">
                      <VALUE>int</VALUE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
  <KEY name="timecreated">
    <VALUE>int</VALUE>
  </KEY>
  <KEY name="usergrade">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="maxgrade">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="gradedby">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="timemodified">
    <VALUE>int</VALUE>
  </KEY>
  <SINGLE>
  </KEY>
  <KEY name="warnings">
    <MULTIPLE>
      <SINGLE>
        <KEY name="item">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </MULTIPLE>
  </KEY>

```



```

</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

gradingform_rubric_grader_gradingpanel_store

Store the grading data for a user from the grader grading panel.

Arguments

component (Required)

The name of the component

General structure

```
string //The name of the component
```

**XML-RPC (PHP structure)**

```
[component] => string
```

REST (POST parameters)

```
component= string
```

contextid (Required)

The ID of the context being graded

General structure

```
int //The ID of the context being graded
```

XML-RPC (PHP structure)

```
[contextid] => int
```

REST (POST parameters)

```
contextid= int
```

itemname (Required)

The grade item itemname being graded

General structure

```
string //The grade item itemname being graded
```

XML-RPC (PHP structure)

```
[itemname] => string
```

REST (POST parameters)

```
itemname= string
```

gradeduserid (Required)

The ID of the user show

General structure

```
int //The ID of the user show
```

XML-RPC (PHP structure)

```
[gradeduserid] => int
```

REST (POST parameters)

```
gradeduserid= int
```

notifyuser (Default to "")

Whetehr to notify the user or not

General structure

```
int Default to "" //Whetehr to notify the user or not
```

XML-RPC (PHP structure)

```
[notifyuser] => int
```

REST (POST parameters)

```
notifyuser= int
```

formdata (Required)

The serialised form data representing the grade

General structure

```
string //The serialised form data representing the grade
```

XML-RPC (PHP structure)

[formdata] => string

REST (POST parameters)

formdata= string

Response**General structure**

```
object {
    templatename string //The template to use when rendering this data
    hasgrade int //Does the user have a grade?
    grade object {
        instanceid int //The id of the current grading instance
        rubricmode string //The mode i.e. evaluate editable
        canedit int //Can the user edit this
        criteria list of (
            object {
                id int //ID of the Criteria
                description string //Description of the Criteria
                remark string Optional //Any remarks for this criterion for the user being assessed
                levels list of (
                    object {
                        id int //ID of level
                        criterionid int //ID of the criterion this matches to
                        score string //What this Level is worth
                        definition string //Definition of the Level
                        checked int //Selected flag
                    }
                ))
            timecreated int //The time that the grade was created
            usergrade string //Current user grade
            maxgrade string //Max possible grade
            gradedby string //The assumed grader of this grading instance
            timemodified int //The time that the grade was last updated
        )
        warnings Optional //List of warnings
        list of (
            //warning
            object {
                item string Optional //item
                itemid int Optional //item id
                warningcode string //the warning code can be used by the client app to implement specific behaviour
                message string //untranslated english message to explain the warning
            }
        )
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [
        [templatename] => string
        [hasgrade] => int
        [grade] =>
            Array
            (
                [
                    [instanceid] => int
                    [rubricmode] => string
                    [canedit] => int
                    [criteria] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [id] => int
                                            [description] => string
                                            [remark] => string
                                            [levels] =>
                                                Array
                                                (
                                                    [
                                                        [0] =>
                                                            Array
                                                            (
                                                                [
                                                                    [id] => int
                                                                    [criterionid] => int
                                                                    [score] => string
                                                                    [definition] => string
                                                                    [checked] => int
                                                                )
                                                            )
                                                    )
                                                )
                                        )
                                    )
                )
            )
        [timecreated] => int
        [usergrade] => string
        [maxgrade] => string
        [gradedby] => string
        [timemodified] => int
    )
)
[warnings] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="templatename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="hasgrade">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="grade">
      <SINGLE>
        <KEY name="instanceid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="rubricmode">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="canedit">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="criteria">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="description">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="remark">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="levels">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="id">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="criterionid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="score">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="definition">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="checked">
                      <VALUE>int</VALUE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
  <KEY name="timecreated">
    <VALUE>int</VALUE>
  </KEY>
  <KEY name="usergrade">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="maxgrade">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="gradedby">
    <VALUE>string</VALUE>
  </KEY>
  <KEY name="timemodified">
    <VALUE>int</VALUE>
  </KEY>
  <SINGLE>
  </KEY>
  <KEY name="warnings">
    <MULTIPLE>
      <SINGLE>
        <KEY name="item">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="itemid">
          <VALUE>int</VALUE>
        </KEY>
      </SINGLE>
    </MULTIPLE>
  </KEY>

```



```

</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

media_videojs_get_language

get language.

Arguments

lang (Required)

language

General structure

string //Language



XML-RPC (PHP structure)

[lang] => string

REST (POST parameters)

lang= string

Response

language pack json

General structure

string //Language pack json

XML-RPC (PHP structure)

```
string
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>string</VALUE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

message_airnotifier_are_notification_preferences_configured

Check if the users have notification preferences configured yet

Arguments

userids (Required)

General structure

```
list of (
int //user ID
)
```

**XML-RPC (PHP structure)**

```
[userids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
userids[0]= int
```

Response

General structure

```
object {
  users //List of preferences by user
  list of (
    object {
      userid int //userid id
      configured int //1 if the user preferences have been configured and 0 if not
    }
  )warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [
    [users] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [userid] => int
                    [configured] => int
                  ]
                )
          )
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  ]
                )
          )
        )
  )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="users">
      <MULTIPLE>
        <SINGLE>
          <KEY name="userid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="configured">
            <VALUE>int</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

message_airnotifier_enable_device

Enables or disables a registered user device so it can receive Push notifications

Arguments

deviceid (Required)

The device id

General structure

```
int //The device id
```

XML-RPC (PHP structure)

```
[deviceid] => int
```

REST (POST parameters)

```
deviceid= int
```

enable (Required)

True for enable the device, false otherwise

General structure

```
int //True for enable the device, false otherwise
```

XML-RPC (PHP structure)

```
[enable] => int
```

REST (POST parameters)

```
enable= int
```

Response**General structure**

```
object {
  success int //True if success
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
  [success] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="success">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**message_airnotifier_get_user_devices**

Return the list of mobile devices that are registered in Moodle for the given user

Arguments**appid** (Required)

App unique id (usually a reversed domain)

General structure

```
string //App unique id (usually a reversed domain)
```

XML-RPC (PHP structure)

```
[appid] => string
```

REST (POST parameters)

appid= string

userid (Default to "0")

User id, 0 for current user

General structure

int Default to "0" //User id, 0 for current user

XML-RPC (PHP structure)

[userid] => int

REST (POST parameters)

userid= int

Response**General structure**

```

object {
  devices  //List of devices
  list of (
    object {
      id int //Device id (in the message_airnotifier table)
      appid string //The app id, something like com.moodle.moodlemobile
      name string //The device name, 'occam' or 'iPhone' etc.
      model string //The device model 'Nexus4' or 'iPad,1' etc.
      platform string //The device platform 'iOS' or 'Android' etc.
      version string //The device version '6.1.2' or '4.2.2' etc.
      pushid string //The device PUSH token/key/identifier/registration id
      uuid string //The device UUID
      enable int //Whether the device is enabled or not
      timecreated int //Time created
      timemodified int //Time modified
    }
  )warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}

```



XML-RPC (PHP structure)

```
Array
(
    [devices] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [appid] => string
                            [name] => string
                            [model] => string
                            [platform] => string
                            [version] => string
                            [pushid] => string
                            [uuid] => string
                            [enable] => int
                            [timecreated] => int
                            [timemodified] => int
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="devices">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="appid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="model">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="platform">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="version">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="pushid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="uuid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="enable">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

message_airnotifier_is_system_configured

Check whether the airnotifier settings have been configured

Arguments

Response

o if the system is not configured, 0 otherwise

General structure

```
int //0 if the system is not configured, 1 otherwise
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



Restricted to logged-in users

Yes

Callable from AJAX

No

message_popup_get_popup_notifications

Retrieve a list of popup notifications for a user

Arguments

useridto (Required)

the user id who received the message, 0 for current user

General structure

```
int //the user id who received the message, 0 for current user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```

newestfirst (Default to "1")

true for ordering by newest first, false for oldest first

General structure

```
int Default to "1" //true for ordering by newest first, false for oldest first
```

XML-RPC (PHP structure)

```
[newestfirst] => int
```

REST (POST parameters)

```
newestfirst= int
```

limit (Default to "0")

the number of results to return

General structure

```
int Default to "0" //the number of results to return
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```



offset (Default to "0")

offset the result set by a given amount

General structure

```
int Default to "0" //offset the result set by a given amount
```

XML-RPC (PHP structure)

```
[offset] => int
```

REST (POST parameters)

```
offset= int
```

Response

```
General structure

object {
  notifications list of (
    //message
    object {
      id int //Notification id (this is not guaranteed to be unique
             //within this result set)
      useridfrom int //User from id
      userido int //User to id
      subject string //The notification subject
      shortenedsubject string //The notification subject shortened
                           //with ellipsis
      text string //The message text formated
      fullmessage string //The message
      fullmessageformat int //fullmessage format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      fullmessagehtml string //The message in html
      smallmessage string //The shorten message
      contexturl string //Context URL
      contexturlname string //Context URL Link name
      timecreated int //Time created
      timecreatedpretty string //Time created in a pretty format
      timeread int //Time read
      read int //notification read status
      deleted int //notification deletion status
      iconurl string //URL for notification icon
      component string Optional //The component that generated the notification
      eventtype string Optional //The type of notification
      customdata string Optional //Custom data to be passed to the message processor.
                           //The data here is serialised using json_encode().
    }
  )
  )unreadcount int //the number of unread message for the given user
}
```

XML-RPC (PHP structure)

```
Array
(
  [notifications] =>
  Array
  (
    [0] =>
    Array
    (
      [
        [id] => int
        [useridfrom] => int
        [userido] => int
        [subject] => string
        [shortenedsubject] => string
        [text] => string
        [fullmessage] => string
        [fullmessageformat] => int
        [fullmessagehtml] => string
        [smallmessage] => string
        [contexturl] => string
        [contexturlname] => string
        [timecreated] => int
        [timecreatedpretty] => string
        [timeread] => int
        [read] => int
        [deleted] => int
        [iconurl] => string
        [component] => string
        [eventtype] => string
        [customdata] => string
      )
    )
  )
  [unreadcount] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="notifications">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="useridfrom">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="useridto">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="shortenedsubject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="text">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullmessage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fullmessageformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullmessagehtml">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="smallmessage">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="contexturl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="contexturlname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecreatedpretty">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timeread">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="read">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="deleted">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="iconurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="component">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="eventtype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="customdata">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="unreadcount">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

message_popup_get_unread_popup_notification_count

Retrieve the count of unread popup notifications for a given user

Arguments

useridto (Required)

the user id who received the message, 0 for any user

General structure

```
int //the user id who received the message, 0 for any user
```

XML-RPC (PHP structure)

```
[useridto] => int
```

REST (POST parameters)

```
useridto= int
```



Response

The count of unread popup notifications

General structure

```
int //The count of unread popup notifications
```

XML-RPC (PHP structure)

```
int
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <VALUE>int</VALUE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_assign_copy_previous_attempt

Copy a students previous attempt to a new attempt.

Arguments

assignmentid (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```



Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

mod_assign_get_assignments

Returns the courses and assignments for the users capability

Arguments

courseids (Default to "Array ()")

0 or more course ids

General structure

```
Default to "Array
(
)
" //0 or more course ids
list of (
int  //course id, empty for retrieving all the courses where the user is enroled in
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [
                [0] => int
            )
        )
```

REST (POST parameters)

```
courseids[0]= int
```

capabilities (Default to "Array ()")

list of capabilities used to filter courses

General structure

```
Default to "Array
(
)
" //List of capabilities used to filter courses
list of (
string  //capability
)
```

XML-RPC (PHP structure)

```
[capabilities] =>
    Array
        (
            [
                [0] => string
            )
        )
```

REST (POST parameters)

```
capabilities[0]= string
```

includenotenrolledcourses (Default to "")

whether to return courses that the user can see even if is not enroled in. This requires the parameter courseids to not be empty.

General structure

```
int  Default to "" //whether to return courses that the user can see
                           even if is not enroled in. This requires the parameter courseids
                           to not be empty.
```

XML-RPC (PHP structure)

```
[includenotenrolledcourses] => int
```

REST (POST parameters)

```
includenotenrolledcourses= int
```

Response

```
General structure

object {
courses //list of courses
list of (
    //course information object
object {
id int //course id
fullname string //course full name
shortname string //course short name
timemodified int //Last time modified
assignments //assignment info
list of (
    //assignment information object
object {
id int //assignment id
cmid int //course module id
course int //course id
name string //assignment name
nosubmissions int //no submissions
submissiondrafts int //submissions drafts
sendnotifications int //send notifications
sendlatenotifications int //send notifications
sendstudentnotifications int //send student notifications (default)
duedate int //assignment due date
allowsubmissionsfromdate int //allow submissions from date
grade int //grade type
timemodified int //last time assignment was modified
completionsubmit int //if enabled, set activity as complete following submission
cutoffdate int //date after which submission is not accepted without an extension
gradingduedate int //the expected date for marking the submissions
teamsubmission int //if enabled, students submit as a team
requireallteammemberssubmit int //if enabled, all team members must submit
teamsubmissiongroupingid int //the grouping id for the team submission groups
blindmarking int //if enabled, hide identities until reveal identities actioned
hidegrader int //If enabled, hide grader to student
revealidentities int //show identities for a blind marking assignment
attemptreopenmethod string //method used to control opening new attempts
maxattempts int //maximum number of attempts allowed
markingworkflow int //enable marking workflow
markingallocation int //enable marking allocation
requiresubmissionstatement int //student must accept submission statement
preventsSubmissionnotinGroup int Optional //Prevent submission not in group
submissionstatement string Optional //Submission statement formatted.
submissionstatementformat int Optional //submissionstatement format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
configs //configuration settings
list of (
    //assignment configuration object
object {
id int Optional //assign_plugin_config id
assignment int Optional //assignment id
plugin string //plugin
subtype string //subtype
name string //name
value string //value
}
)intro string Optional //assignment intro, not always returned because it depends on the activity config.
introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction text
list of (
    //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)introattachments Optional //intro attachments files
list of (
    //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
```

```
}

)activity string  Optional //Description of activity
activityformat int  Optional //activity format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
activityattachments  Optional //Files from activity field
list of (
    //File.
object {
filename string  Optional //File name.
filepath string  Optional //File path.
filesize int  Optional //File size.
fileurl string  Optional //Downloadable file url.
timemodified int  Optional //Time modified.
mimetype string  Optional //File mime type.
isexternalfile int  Optional //Whether is an external file.
repositorytype string  Optional //The repository type for external files.
}
)timelimit int  Optional //Time limit to complete assignment
submissionattachments int  Optional //Flag to only show files during submission
}
)
)warnings  Optional //List of warnings
list of (
    //warning
object {
item string  Optional //item can be 'course' (errorcode 1 or 2) or 'module' (errorcode 1)
itemid int  Optional //When item is a course then itemid is a course id. When the item is a module then itemid is a module id
warningcode string  //errorcode can be 1 (no access rights) or 2 (not enrolled or no permissions)
message string  //untranslated english message to explain the warning
}
)
}
```



XML-RPC (PHP structure)

```

Array
(
    [
        [courses] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [fullname] => string
                                [shortname] => string
                                [timemodified] => int
                                [assignments] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [id] => int
                                                        [cmid] => int
                                                        [course] => int
                                                        [name] => string
                                                        [nosubmissions] => int
                                                        [submissiondrafts] => int
                                                        [sendnotifications] => int
                                                        [sendlatenotifications] => int
                                                        [sendstudentnotifications] => int
                                                        [duedate] => int
                                                        [allowsubmissionsfromdate] => int
                                                        [grade] => int
                                                        [timemodified] => int
                                                        [completionsubmit] => int
                                                        [cutoffdate] => int
                                                        [gradingduedate] => int
                                                        [teamsubmission] => int
                                                        [requireallteammemberssubmit] => int
                                                        [teamsubmissiongroupingid] => int
                                                        [blindmarking] => int
                                                        [hidegrader] => int
                                                        [revealedentities] => int
                                                        [attemptreopenmethod] => string
                                                        [maxattempts] => int
                                                        [markingworkflow] => int
                                                        [markingallocation] => int
                                                        [requiresubmissionstatement] => int
                                                        [preventsubmissionnotingroup] => int
                                                        [submissionstatement] => string
                                                        [submissionstatementformat] => int
                                                        [configs] =>
                                                            Array
                                                            (
                                                                [
                                                                    [0] =>
                                                                        Array
                                                                        (
                                                                            [
                                                                                [id] => int
                                                                                [assignment] => int
                                                                                [plugin] => string
                                                                                [subtype] => string
                                                                                [name] => string
                                                                                [value] => string
                                                                            )
                                                                )
                                                            )
                                                        )
                                                    )
                                                )
                                            [intro] => string
                                            [introformat] => int
                                            [introfiles] =>
                                                Array
                                                (
                                                    [
                                                        [0] =>
                                                            Array
                                                            (
                                                                [
                                                                    [filename] => string
                                                                    [filepath] => string
                                                                    [filesize] => int
                                                                    [fileurl] => string
                                                                    [timemodified] => int
                                                                    [mimetype] => string
                                                                    [isexternalfile] => int
                                                                    [repositorytype] => string
                                                                )
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)

```



```
[introattachments] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [filename] => string
                [filepath] => string
                [filesize] => int
                [fileurl] => string
                [timemodified] => int
                [mimetype] => string
                [isexternalfile] => int
                [repositorytype] => string
              )
            )
      ]
    )
  [
    [activity] => string
    [activityformat] => int
    [activityattachments] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [filename] => string
                    [filepath] => string
                    [filesize] => int
                    [fileurl] => string
                    [timemodified] => int
                    [mimetype] => string
                    [isexternalfile] => int
                    [repositorytype] => string
                  )
                )
            ]
          )
        [
          [timelimit] => int
          [submissionattachments] => int
        )
      )
  )
)
[
  [
    [
      [
        [
          [
            [
              [
                [
                  [
                    [
                      [
                        [
                          [
                            [
                              [
                                [
                                  [
                                    [
                                      [
                                        [
                                          [
                                            [
                                              [
                                                [
                                                  [
                                                    [
                                                      [
                                                        [
                                                          [
                                                            [
                                                              [
                                                                [
                                                                  [
                                                                    [
                                                                      [
                                                                        [
                                                                          [
                                                                            [
                                                                              [
                                                                                [
                                                                                  [
                                                                                    [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
................................................................
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="courses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="shortname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                <KEY name="assignments">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="id">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="cmid">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="course">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="name">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="nosubmissions">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="submissiondrafts">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="sendnotifications">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="sendlatenotifications">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="sendstudentnotifications">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="duedate">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="allowsubmissionsfromdate">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="grade">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="timemodified">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="completionsubmit">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="cutoffdate">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="gradingduedate">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="teamsubmission">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="requireallteammemberssubmit">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="teamsubmissiongroupingid">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="blindmarking">
                                <VALUE>int</VALUE>
                            </KEY>
                        </SINGLE>
                    </MULTIPLE>
                </KEY>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="hidegrader">
    <VALUE>int</VALUE>
</KEY>
<KEY name="revealidentities">
    <VALUE>int</VALUE>
</KEY>
<KEY name="attemptreopenmethod">
    <VALUE>string</VALUE>
</KEY>
<KEY name="maxattempts">
    <VALUE>int</VALUE>
</KEY>
<KEY name="markingworkflow">
    <VALUE>int</VALUE>
</KEY>
<KEY name="markingallocation">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requiresubmissionstatement">
    <VALUE>int</VALUE>
</KEY>
<KEY name="preventsSubmissionnotinGroup">
    <VALUE>int</VALUE>
</KEY>
<KEY name="submissionstatement">
    <VALUE>string</VALUE>
</KEY>
<KEY name="submissionstatementformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="configs">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="assignment">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="plugin">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="subtype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="intro">
    <VALUE>string</VALUE>
</KEY>
<KEY name="introformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="introfiles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="introattachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="activity">
    <VALUE>string</VALUE>
</KEY>
<KEY name="activityformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="activityattachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="timelimit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="submissionattachments">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
```



```

        </MULTIPLE>
    </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_get_grades

Returns grades from the assignment



Arguments

assignmentids (Required)

1 or more assignment ids

General structure

```

//1 or more assignment ids
list of (
int //assignment id
)

```

XML-RPC (PHP structure)

```

[assignmentids] =>
Array
(
    [0] => int
)

```

REST (POST parameters)

```
assignmentids[0]= int
```

since (Default to "0")

timestamp, only return records where timemodified >= since

General structure

```
int Default to "0" //timestamp, only return records where timemodified >= since
```

XML-RPC (PHP structure)

```
[since] => int
```

REST (POST parameters)

```
since= int
```

Response**General structure**

```
object {
  assignments //list of assignment grade information
  list of (
    object {
      assignmentid int //assignment id
      grades list of (
        //grade information
        object {
          id int //grade id
          assignment int Optional //assignment id
          userid int //student id
          attemptnumber int //attempt number
          timecreated int //grade creation time
          timemodified int //grade last modified time
          grader int //grader, -1 if grader is hidden
          grade string //grade
          gradefordisplay string Optional //grade rendered into a format suitable for display
        }
      )
    }
  )
  warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item is always 'assignment'
      itemid int Optional //when errorcode is 3 then itemid is an assignment id. When errorcode is 1, itemid is a course module id
      warningcode string //errorcode can be 3 (no grades found) or 1 (no permission to get grades)
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
    [assignments] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [assignmentid] => int
                            [grades] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [
                                                        [id] => int
                                                        [assignment] => int
                                                        [userid] => int
                                                        [attemptnumber] => int
                                                        [timecreated] => int
                                                        [timemodified] => int
                                                        [grader] => int
                                                        [grade] => string
                                                        [gradefordisplay] => string
                                                    )
                                                )
                                        )
                                    )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="assignments">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="assignmentid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="grades">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="assignment">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="attemptnumber">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="grader">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="grade">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="gradefordisplay">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_get_participant

Get a participant for an assignment, with some summary info about their submissions.

Arguments**assignid** (Required)

assign instance id

General structure

```
int //assign instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

REST (POST parameters)

```
assignid= int
```

userid (Required)

user id

General structure

```
int //user id
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

embeduser (Default to "")

user id

General structure

```
int Default to "" //user id
```

XML-RPC (PHP structure)

```
[embeduser] => int
```

REST (POST parameters)

```
embeduser= int
```

Response**General structure**

```
object {
    id int //ID of the user
    fullname string //The fullname of the user
    submitted int //have they submitted their assignment
    requiregrading int //is their submission waiting for grading
    grantedextension int //have they been granted an extension
    blindmarking int //is blind marking enabled for this assignment
    allowsubmissionsfromdate int //allowsubmissionsfromdate for the user
    duedate int //duedate for the user
    cutoffdate int //cutoffdate for the user
    dueDateString string //duedate for the user
    groupid int Optional //for group assignments this is the group id
    groupname string Optional //for group assignments this is the group name
    submissionstatus string Optional //The submission status (new, draft, reopened or submitted).
        Empty when not submitted.

    user object {
        id int //ID of the user
        username string Optional //The username
        firstname string Optional //The first name(s) of the user
        lastname string Optional //The family name of the user
        fullname string //The fullname of the user
        email string Optional //An email address - allow email as root@localhost
        address string Optional //Postal address
        phone1 string Optional //Phone 1
        phone2 string Optional //Phone 2
        department string Optional //department
        institution string Optional //institution
        idnumber string Optional //An arbitrary ID code number perhaps from the institution
        interests string Optional //user interests (separated by commas)
        firstaccess int Optional //first access to the site (0 if never)
        lastaccess int Optional //last access to the site (0 if never)
        auth string Optional //Auth plugins include manual, Ldap, etc
        suspended int Optional //Suspend user account, either false to enable user login or true to disable it
        confirmed int Optional //Active user: 1 if confirmed, 0 otherwise
        lang string Optional //Language code such as "en", must exist on server
        calendartype string Optional //Calendar type such as "gregorian", must exist on server
        theme string Optional //Theme name such as "standard", must exist on server
        timezone string Optional //Timezone code such as Australia/Perth, or 99 for default
        mailformat int Optional //Mail format code is 0 for plain text, 1 for HTML etc
        description string Optional //User profile description
        descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
        city string Optional //Home city of the user
        country string Optional //Home country code of the user, such as AU or CZ
        profileimageurlsmall string //User image profile URL - small version
        profileimageurl string //User image profile URL - big version
        customfields Optional //User custom fields (also known as user profile fields)
        list of (
            object {
                type string //The type of the custom field - text field, checkbox...
                value string //The value of the custom field (as stored in the database)
                displayvalue string Optional //The value of the custom field for display
                name string //The name of the custom field
                shortname string //The shortname of the custom field - to be able to build the field class in the code
            }
        )preferences Optional //Users preferences
        list of (
            object {
                name string //The name of the preferences
                value string //The value of the preference
            }
        )
    }
}
```

XML-RPC (PHP structure)

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="fullname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="submitted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="requiregrading">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="grantedextension">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="blindmarking">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="allowsubmissionsfromdate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="cutoffdate">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="duedatestr">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="groupid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="groupname">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="submissionstatus">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="user">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="username">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="firstname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="lastname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fullname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="email">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="address">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone1">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="phone2">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="department">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="institution">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="idnumber">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="interests">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="firstaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lastaccess">
    <VALUE>int</VALUE>
</KEY>
<KEY name="auth">
    <VALUE>string</VALUE>
</KEY>
<KEY name="suspended">
    <VALUE>int</VALUE>
</KEY>
<KEY name="confirmed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lang">
    <VALUE>string</VALUE>
</KEY>
<KEY name="calendartype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="theme">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timezone">
    <VALUE>string</VALUE>
</KEY>
<KEY name="mailformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="description">
    <VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="city">
    <VALUE>string</VALUE>
</KEY>
<KEY name="country">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
    <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="customfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</SINGLE>
</KEY>
</SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_assign_get_submission_status

Returns information about an assignment submission status for a given user.

Arguments**assignid (Required)**

assignment instance id

General structure

```
int //assignment instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

**REST (POST parameters)**

```
assignid= int
```

userid (Default to "0")

user id (empty for current user)

General structure

```
int Default to "0" //user id (empty for current user)
```

XML-RPC (PHP structure)

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

groupid (Default to "0")

filter by users in group (used for generating the grading summary). 0 for all groups information, any other empty value will calculate current group.

General structure

```
int Default to "0" //filter by users in group (used for generating the grading summary).
0 for all groups information, any other empty value will calculate current group.
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response



```
General structure

object {
gradingsummary Optional //Grading information.
object {
participantcount int //Number of users who can submit.
submissiondraftscount int //Number of submissions in draft status.
submissionsenabled int //Whether submissions are enabled or not.
submissionssubmittedcount int //Number of submissions in submitted status.
submissionsneedgradingcount int //Number of submissions that need grading.
warnofungroupedusers string //Whether we need to warn people that there
} are users without groups ('warningrequired'), warn
      people there are users who will submit in the default
      group ('warningoptional') or no warning ('').
lastattempt Optional //Last attempt information.
object {
submission Optional //submission info
object {
id int //submission id
userid int //student id
attemptnumber int //attempt number
timecreated int //submission creation time
timemodified int //submission last modified time
timestarted int Optional //submission start time
status string //submission status
groupid int //group id
assignment int Optional //assignment id
latest int Optional //latest attempt
plugins Optional //plugins
list of (
object {
type string //submission plugin type
name string //submission plugin name
fileareas Optional //fileareas
list of (
object {
area string //file area
files Optional //files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)}
)editorfields Optional //editorfields
list of (
object {
name string //field name
description string //field description
text string //field value
format int //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)}
)gradingstatus string Optional //Grading status.
}
teamsubmission Optional //submission info
object {
id int //submission id
userid int //student id
attemptnumber int //attempt number
timecreated int //submission creation time
timemodified int //submission last modified time
timestarted int Optional //submission start time
status string //submission status
groupid int //group id
assignment int Optional //assignment id
latest int Optional //latest attempt
plugins Optional //plugins
list of (
object {
type string //submission plugin type
name string //submission plugin name
fileareas Optional //fileareas
list of (
object {
}
```

```
area string //file area
files Optional //files
list of (
  //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)}
)editorfields Optional //editorfields
list of (
object {
name string //field name
description string //field description
text string //field value
format int //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)}
)gradingstatus string Optional //Grading status.
)
submissiongroup int Optional //The submission group id (for group submissions only).
submissiongroupmemberswhoneedtositin Optional //List of users who still need to submit (for group submissions only).
list of (
int //USER id.
)submissionsenabled int //Whether submissions are enabled or not.
locked int //Whether new submissions are Locked.
graded int //Whether the submission is graded.
canedit int //Whether the user can edit the current submission.
caneditowner int //Whether the owner of the submission can edit it.
cansubmit int //Whether the user can submit.
extensionduedate int //Extension due date.
timelimit int Optional //Time limit for submission.
blindmarking int //Whether blind marking is enabled.
gradingstatus string //Grading status.
usergroups //User groups in the course.
list of (
int //Group id.
)}
feedback Optional //Feedback for the last attempt.
object {
grade Optional //grade information
object {
id int //grade id
assignment int Optional //assignment id
userid int //student id
attemptnumber int //attempt number
timecreated int //grade creation time
timemodified int //grade last modified time
grader int //grader, -1 if grader is hidden
grade string //grade
gradefordisplay string Optional //Grade rendered into a format suitable for display
}
gradefordisplay string //Grade rendered into a format suitable for display.
gradeddate int //The date the user was graded.
plugins Optional //Plugins info.
list of (
object {
type string //submission plugin type
name string //submission plugin name
fileareas Optional //fileareas
list of (
object {
area string //file area
files Optional //files
list of (
  //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
)}}}
```



```
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
)
)editorfields Optional //editorfields
list of (
object {
name string //field name
description string //field description
text string //field value
format int //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
)
)
)assignmentdata Optional //Extra information about assignment
object {
attachments Optional //Intro and activity attachments
object {
intro Optional //Intro attachments files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
activity Optional //Activity attachments files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
)
activity string Optional //Text of activity
activityformat int Optional //activity format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
)
```



XML-RPC (PHP structure)

```
[id] => int
[userid] => int
[attemptnumber] => int
[timecreated] => int
[timemodified] => int
[timestarted] => int
[status] => string
[groupid] => int
[assignment] => int
[latest] => int
[plugins] =>
    Array
        (
            [
                [0] =>
                    Array
                        (
                            [
                                [type] => string
                                [name] => string
                                [fileareas] =>
                                    Array
                                        (
                                            [
                                                [0] =>
                                                    Array
                                                        (
                                                            [
                                                                [area] => string
                                                                [files] =>
                                                                    Array
                                                                        (
                                                                            [
                                                                                [0] =>
                                                                                    Array
                                                                                        (
                                                                                            [
                                                                                                [filename] => string
                                                                                                [filepath] => string
                                                                                                [filesize] => int
                                                                                                [fileurl] => string
                                                                                                [timemodified] => int
                                                                                                [mimetype] => string
                                                                                                [isexternalfile] => int
                                                                                                [repositorytype] => string
                                                                                            )
                                                                                        )
                                                                )
                                                            )
                                                        )
                                                    )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
[gradingstatus] => string
)
[submissongroup] => int
[submissongroupmemberswhoneedtosit] =>
    Array
        (
            [
                [0] => int
            )
        )
[submissionsenabled] => int
[locked] => int
[graded] => int
[canedit] => int
[caneditowner] => int
[cansubmit] => int
[extensionduedate] => int
[timelimit] => int
[blindmarking] => int
[gradingstatus] => string
[usergroups] =>
    Array
        (
            [
                [0] => int
            )
        )
)
[feedback] =>
    Array
```




^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="gradingsummary">
            <SINGLE>
                <KEY name="participantcount">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="submissiondraftscount">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="submissionsenabled">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="submissionssubmittedcount">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="submissionsneedgradingcount">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warnofungroupedusers">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="lastattempt">
            <SINGLE>
                <KEY name="submission">
                    <SINGLE>
                        <KEY name="id">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="userid">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="attemptnumber">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="timecreated">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="timemodified">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="timestarted">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="status">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="groupid">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="assignment">
                            <VALUE>int</VALUE>
                        </KEY>
                        <KEY name="latest">
                            <VALUE>int</VALUE>
                        </KEY>
                    </SINGLE>
                <KEY name="plugins">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="type">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="name">
                                <VALUE>string</VALUE>
                            </KEY>
                        <KEY name="fileareas">
                            <MULTIPLE>
                                <SINGLE>
                                    <KEY name="area">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="files">
                                        <MULTIPLE>
                                            <SINGLE>
                                                <KEY name="filename">
                                                    <VALUE>string</VALUE>
                                                </KEY>
                                                <KEY name="filepath">
```



```

        <VALUE>string</VALUE>
    </KEY>
    <KEY name="filesize">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="fileurl">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="timemodified">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="mimetype">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="isexternalfile">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="repositorytype">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="editorfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="format">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="gradingstatus">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="teamsubmission">
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="attemptnumber">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timestarted">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="status">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="groupid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="assignment">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="latest">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>

```



```
</KEY>
<KEY name="plugins">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="fileareas">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="area">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="files">
                            <MULTIPLE>
                                <SINGLE>
                                    <KEY name="filename">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="filepath">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="filesize">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="fileurl">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="timemodified">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="mimetype">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="isexternalfile">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="repositorytype">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                </SINGLE>
                            </MULTIPLE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="editorfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="format">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
<KEY name="gradingstatus">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="submissiongroup">
    <VALUE>int</VALUE>
</KEY>
<KEY name="submissiongroupmemberswhoneedtosubmit">
    <MULTIPLE>
        <VALUE>int</VALUE>
    </MULTIPLE>
</KEY>
</KEY>
```



```
<KEY name="submissionsenabled">
    <VALUE>int</VALUE>
</KEY>
<KEY name="locked">
    <VALUE>int</VALUE>
</KEY>
<KEY name="graded">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canedit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="caneditowner">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cansubmit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="extensionduedate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timelimit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="blindmarking">
    <VALUE>int</VALUE>
</KEY>
<KEY name="gradingstatus">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usergroups">
    <MULTIPLE>
        <VALUE>int</VALUE>
    </MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="feedback">
    <SINGLE>
        <KEY name="grade">
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="assignment">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="attemptnumber">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timecreated">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="grader">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="grade">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="gradefordisplay">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="gradefordisplay">
    <VALUE>string</VALUE>
</KEY>
<KEY name="gradeddate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="plugins">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="fileareas">
<MULTIPLE>
<SINGLE>
<KEY name="area">
<VALUE>string</VALUE>
</KEY>
<KEY name="files">
<MULTIPLE>
<SINGLE>
<KEY name="filename">
<VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
<VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
<VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
<VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
<VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="editorfields">
<MULTIPLE>
<SINGLE>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="text">
<VALUE>string</VALUE>
</KEY>
<KEY name="format">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="previousattempts">
<MULTIPLE>
<SINGLE>
<KEY name="attemptnumber">
<VALUE>int</VALUE>
</KEY>
<KEY name="submission">
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="attemptnumber">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
```



```
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timestarted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="status">
    <VALUE>string</VALUE>
</KEY>
<KEY name="groupid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="assignment">
    <VALUE>int</VALUE>
</KEY>
<KEY name="latest">
    <VALUE>int</VALUE>
</KEY>
<KEY name="plugins">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="fileareas">
                <MULTIPLE>
                    <SINGLE>
                        <KEY name="area">
                            <VALUE>string</VALUE>
                        </KEY>
                        <KEY name="files">
                            <MULTIPLE>
                                <SINGLE>
                                    <KEY name="filename">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="filepath">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="filesize">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="fileurl">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="timemodified">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="mimetype">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="isexternalfile">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="repositorytype">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                </SINGLE>
                            </MULTIPLE>
                        </KEY>
                    </SINGLE>
                </MULTIPLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="editorfields">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="text">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="format">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="gradingstatus">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="grade">
<SINGLE>
    <KEY name="id">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="assignment">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="userid">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="attemptsnumber">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="grader">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="grade">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="gradefordisplay">
        <VALUE>string</VALUE>
    </KEY>
</SINGLE>
</KEY>
<KEY name="feedbackplugins">
<MULTIPLE>
<SINGLE>
    <KEY name="type">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="name">
        <VALUE>string</VALUE>
    </KEY>
    <KEY name="fileareas">
<MULTIPLE>
<SINGLE>
        <KEY name="area">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="files">
<MULTIPLE>
<SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
```



```
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
<KEY name="editorfields">
<MULTIPLE>
<SINGLE>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="text">
<VALUE>string</VALUE>
</KEY>
<KEY name="format">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
<KEY name="assignmentdata">
<SINGLE>
<KEY name="attachments">
<SINGLE>
<KEY name="intro">
<MULTIPLE>
<SINGLE>
<KEY name="filename">
<VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
<VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
<VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
<VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
<VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="activity">
<MULTIPLE>
<SINGLE>
<KEY name="filename">
<VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
<VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
<VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
```



```

        <VALUE>string</VALUE>
    </KEY>
    <KEY name="isexternalfile">
        <VALUE>int</VALUE>
    </KEY>
    <KEY name="repositorytype">
        <VALUE>string</VALUE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="activity">
    <VALUE>string</VALUE>
</KEY>
<KEY name="activityformat">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_get_submissions

Returns the submissions for assignments

Arguments

assignmentids (Required)

1 or more assignment ids

General structure

```
//1 or more assignment ids  
list of (  
int //assignment id  
)
```

XML-RPC (PHP structure)

```
[assignmentids] =>  
    Array  
        (  
            [0] => int  
        )
```

REST (POST parameters)

```
assignmentids[0]= int
```

status (Default to "")

status

General structure

```
string Default to "" //status
```

XML-RPC (PHP structure)

```
[status] => string
```

REST (POST parameters)

```
status= string
```

since (Default to "0")

submitted since

General structure

```
int Default to "0" //submitted since
```

**XML-RPC (PHP structure)**

```
[since] => int
```

REST (POST parameters)

```
since= int
```

before (Default to "0")

submitted before

General structure

```
int Default to "0" //submitted before
```

XML-RPC (PHP structure)

```
[before] => int
```

REST (POST parameters)

before= int

Response

General structure

```

object {
assignments //assignment submissions
list of (
object {
assignmentid int //assignment id
submissions list of (
//submission info
object {
id int //submission id
userid int //student id
attemptnumber int //attempt number
timecreated int //submission creation time
timemodified int //submission last modified time
timestarted int Optional //submission start time
status string //submission status
groupid int //group id
assignment int Optional //assignment id
latest int Optional //latest attempt
plugins Optional //plugins
list of (
object {
type string //submission plugin type
name string //submission plugin name
fileareas Optional //fileareas
list of (
object {
area string //file area
files Optional //files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
)
editorfields Optional //editorfields
list of (
object {
name string //field name
description string //field description
text string //field value
format int //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
)
gradingstatus string Optional //Grading status.
)
)
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
}

```



XML-RPC (PHP structure)

```
[0] =>
    Array
    (
        [item] => string
        [itemid] => int
        [warningcode] => string
        [message] => string
    )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="assignments">
      <MULTIPLE>
        <SINGLE>
          <KEY name="assignmentid">
            <VALUE>int</VALUE>
          </KEY>
        <KEY name="submissions">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="userid">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="attemptnumber">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="timecreated">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="timestarted">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="status">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="groupid">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="assignment">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="latest">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="plugins">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="type">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="fileareas">
                      <MULTIPLE>
                        <SINGLE>
                          <KEY name="area">
                            <VALUE>string</VALUE>
                          </KEY>
                        <KEY name="files">
                          <MULTIPLE>
                            <SINGLE>
                              <KEY name="filename">
                                <VALUE>string</VALUE>
                              </KEY>
                              <KEY name="filepath">
                                <VALUE>string</VALUE>
                              </KEY>
                              <KEY name="filesize">
                                <VALUE>int</VALUE>
                              </KEY>
                              <KEY name="fileurl">
                                <VALUE>string</VALUE>
                              </KEY>
                              <KEY name="timemodified">
                                <VALUE>int</VALUE>
                              </KEY>
                              <KEY name="mimetype">
                                <VALUE>string</VALUE>
                              </KEY>
                              <KEY name="isexternalfile">
                                <VALUE>int</VALUE>
                              </KEY>
                            </SINGLE>
                          </MULTIPLE>
                        </SINGLE>
                      </MULTIPLE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
      </SINGLE>
    </MULTIPLE>
  </KEY>
</RESPONSE>
```

```

<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</MULTIPLE>
<KEY>
    <KEY name="editorfields">
        <MULTIPLE>
            <SINGLE>
                <KEY name="name">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="description">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="text">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="format">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="gradingstatus">
        <VALUE>string</VALUE>
    </KEY>
    <SINGLE>
        <MULTIPLE>
            <KEY>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_get_user_flags

Returns the user flags for assignments

Arguments**assignmentids** (Required)

1 or more assignment ids

General structure

```
//1 or more assignment ids
list of (
int //assignment id
)
```

XML-RPC (PHP structure)

```
[assignmentids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

assignmentids[0]= int

Response**General structure**

```
object {
assignments //List of assign user flag information
list of (
object {
assignmentid int //assignment id
userflags list of (
object {
id int //user flag id
userid int //student id
locked int //Locked
mailed int //mailed
extensionduedate int //extension due date
workflowstate string Optional //marking workflow state
allocatedmarker int //allocated marker
})
})
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item is always 'assignment'
itemid int Optional //when errorcode is 3 then itemid is an assignment id. When errorcode is 1, itemid is a course module id
warningcode string //errorcode can be 3 (no user flags found) or 1 (no permission to get user flags)
message string //untranslated english message to explain the warning
})
})
```

XML-RPC (PHP structure)

```
Array
(
    [assignments] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [assignmentid] => int
                            [userflags] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [
                                                       id] => int
                                                       [userid] => int
                                                       [locked] => int
                                                       [mailed] => int
                                                       [extensionduedate] => int
                                                       [workflowstate] => string
                                                       [allocatedmarker] => int
                                                )
                                    )
                        )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="assignments">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="assignmentid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userflags">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="locked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mailed">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="extensionduedate">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="workflowstate">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="allocatedmarker">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_get_user_mappings

Returns the blind marking mappings for assignments

Arguments**assignmentids** (Required)

1 or more assignmentids

General structure

```
//1 or more assignment ids
list of (
int //assignment id
)
```

XML-RPC (PHP structure)

```
[assignmentids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

assignmentids[0]= int

Response**General structure**

```
object {
assignments //List of assign user mapping data
list of (
object {
assignmentid int //assignment id
mappings list of (
object {
id int //user mapping id
userid int //student id
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item is always 'assignment'
itemid int Optional //when errorcode is 3 then itemid is an assignment id. When errorcode is 1, itemid is a course module id
warningcode string //errorcode can be 3 (no user mappings found) or 1 (no permission to get user mappings)
message string //untranslated english message to explain the warning
})
)
}
```

XML-RPC (PHP structure)

```
Array
(
    [assignments] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [assignmentid] => int
                            [mappings] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [userid] => int
                                                )
                                        )
                                    )
                )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="assignments">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="assignmentid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mappings">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_list_participants

List the participants for a single assignment, with some summary info about their submissions.

Arguments**assignid** (Required)

assign instance id

General structure

```
int //assign instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

REST (POST parameters)

```
assignid= int
```

groupid (Required)

group id

General structure

```
int //group id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

filter (Required)

search string to filter the results

General structure

```
string //search string to filter the results
```

XML-RPC (PHP structure)

```
[filter] => string
```

**REST (POST parameters)**

```
filter= string
```

skip (Default to "0")

number of records to skip

General structure

```
int Default to "0" //number of records to skip
```

XML-RPC (PHP structure)

```
[skip] => int
```

REST (POST parameters)

skip= int

limit (Default to "0")

maximum number of records to return

General structureint **Default to "0"** //maximum number of records to return**XML-RPC (PHP structure)**

[limit] => int

REST (POST parameters)

limit= int

onlyids (Default to "")

Do not return all user fields

General structureint **Default to ""** //Do not return all user fields**XML-RPC (PHP structure)**

[onlyids] => int

REST (POST parameters)

onlyids= int

includeenrolments (Default to "1")

Do return courses where the user is enrolled

General structureint **Default to "1"** //Do return courses where the user is enrolled**XML-RPC (PHP structure)**

[includeenrolments] => int

REST (POST parameters)

includeenrolments= int

tablesort (Default to "")

Apply current user table sorting preferences.

General structureint **Default to ""** //Apply current user table sorting preferences.**XML-RPC (PHP structure)**

[tablesort] => int

REST (POST parameters)

tablesort= int

Response

^

```

General structure

list of (
object {
id int //ID of the user
username string Optional //The username
firstname string Optional //The first name(s) of the user
lastname string Optional //The family name of the user
fullname string //The fullname of the user
email string Optional //Email address
address string Optional //Postal address
phone1 string Optional //Phone 1
phone2 string Optional //Phone 2
department string Optional //department
institution string Optional //institution
idnumber string Optional //The idnumber of the user
interests string Optional //user interests (separated by commas)
firstaccess int Optional //first access to the site (0 if never)
lastaccess int Optional //last access to the site (0 if never)
suspended int Optional //Suspend user account, either false to enable user login or true to disable it
description string Optional //User profile description
descriptionformat int Optional //int format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
city string Optional //Home city of the user
country string Optional //Home country code of the user, such as AU or CZ
profileimageurlsmall string Optional //User image profile URL - small version
profileimageurl string Optional //User image profile URL - big version
customfields Optional //User custom fields (also known as user profile fields)
list of (
object {
type string //The type of the custom field - text field, checkbox...
value string //The value of the custom field (as stored in the database)
displayvalue string Optional //The value of the custom field for display
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
})
preferences Optional //Users preferences
list of (
object {
name string //The name of the preferences
value string //The value of the preference
})
recordid int //record id
groups Optional //user groups
list of (
object {
id int //group id
name string //group name
description string //group description
})
roles Optional //user roles
list of (
object {
roleid int //role id
name string //role name
shortname string //role shortname
sortorder int //role sortorder
})
enrolledcourses Optional //Courses where the user is enrolled - Limited by which courses the user is able to see
list of (
object {
id int //Id of the course
fullname string //Fullname of the course
shortname string //Shortname of the course
})
submitted int //have they submitted their assignment
requiregrading int //is their submission waiting for grading
grantedextension int //have they been granted an extension
submissionstatus string Optional //The submission status (new, draft, reopened or submitted).
Empty when not submitted.
groupid int Optional //for group assignments this is the group id
groupname string Optional //for group assignments this is the group name
})
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
        (
            [id] => int
            [username] => string
            [firstname] => string
            [lastname] => string
            [fullname] => string
            [email] => string
            [address] => string
            [phone1] => string
            [phone2] => string
            [department] => string
            [institution] => string
            [idnumber] => string
            [interests] => string
            [firstaccess] => int
            [lastaccess] => int
            [suspended] => int
            [description] => string
            [descriptionformat] => int
            [city] => string
            [country] => string
            [profileimageurlsmall] => string
            [profileimageurl] => string
            [customfields] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [type] => string
                            [value] => string
                            [displayvalue] => string
                            [name] => string
                            [shortname] => string
                        )
                )
            [preferences] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [name] => string
                            [value] => string
                        )
                )
            [recordid] => int
            [groups] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [id] => int
                            [name] => string
                            [description] => string
                        )
                )
            [roles] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [roleid] => int
                            [name] => string
                            [shortname] => string
                            [sortorder] => int
                        )
                )
            [enrolledcourses] =>
                Array
                (
                    [0] =>
                        Array
                        (

```



```
        [id] => int
        [fullname] => string
        [shortname] => string
    )
)
[submitted] => int
[requiregrading] => int
[grantedextension] => int
[submissionstatus] => string
[groupid] => int
[groupname] => string
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="username">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="lastname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="fullname">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="email">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="address">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone1">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="phone2">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="department">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="institution">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="idnumber">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="interests">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="firstaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="lastaccess">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="suspended">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="description">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="descriptionformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="city">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="country">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurlsmall">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="profileimageurl">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="customfields">
        <MULTIPLE>
          <SINGLE>
            <KEY name="type">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="displayvalue">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```



```
</KEY>
<KEY name="name">
    <VALUE>string</VALUE>
</KEY>
<KEY name="shortname">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="preferences">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="value">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
<KEY name="recordid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groups">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="description">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
<KEY name="roles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="roleid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="sortorder">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
</KEY>
<KEY name="enrolledcourses">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="shortname">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
<KEY name="submitted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="requiregrading">
    <VALUE>int</VALUE>
</KEY>
<KEY name="grantedextension">
    <VALUE>int</VALUE>
</KEY>
<KEY name="submissionstatus">
```



```

<VALUE>string</VALUE>
</KEY>
<KEY name="groupid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="groupname">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_assign_lock_submissions

Prevent students from making changes to a list of submissions

Arguments

assignmentid (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```



XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userids (Required)

1 or more user ids

General structure

```
//1 or more user ids
list of (
int //user id
)
```

XML-RPC (PHP structure)

```
[userids] =>
    Array
    (
        [
            [0] => int
        )
    )
```

REST (POST parameters)

```
userids[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
object {
    item string  Optional //item
    itemid int  Optional //item id
    warningcode string  //the warning code can be used by the client app to implement specific behaviour
    message string  //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_reveal_identities

Reveal the identities for a blind marking assignment

Arguments**assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

Response

list of warnings

**General structure**

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
        Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

mod_assign_revert_submissions_to_draft

Reverts the list of submissions to draft status

Arguments**assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userids (Required)

1 or more user ids

General structure

```
//1 or more user ids
list of (
int //user id
)
```

XML-RPC (PHP structure)

```
[userids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
userids[0]= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_save_grade

Save a grade update for a single student.

**Arguments****assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userid (Required)

The student id to operate on

General structure

```
int //The student id to operate on
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

grade (Required)

The new grade for this user. Ignored if advanced grading used

General structure

```
double //The new grade for this user. Ignored if advanced grading used
```

XML-RPC (PHP structure)

```
[grade] => double
```

REST (POST parameters)

```
grade= double
```

attemptnumber (Required)

The attempt number (-1 means latest attempt)

General structure

```
int //The attempt number (-1 means latest attempt)
```

XML-RPC (PHP structure)

```
[attemptnumber] => int
```

REST (POST parameters)

```
attemptnumber= int
```

**addattempt** (Required)

Allow another attempt if the attempt reopen method is manual

General structure

```
int //Allow another attempt if the attempt reopen method is manual
```

XML-RPC (PHP structure)

```
[addattempt] => int
```

REST (POST parameters)

```
addattempt= int
```

workflowstate (Required)

The next marking workflow state

General structure

```
string //The next marking workflow state
```

XML-RPC (PHP structure)

```
[workflowstate] => string
```

REST (POST parameters)

```
workflowstate= string
```

applytoall (Required)

If true, this grade will be applied to all members of the group (for group assignments).

General structure

```
int //If true, this grade will be applied to all members of the group (for group assignments).
```

XML-RPC (PHP structure)

```
[applytoall] => int
```

REST (POST parameters)

```
applytoall= int
```

plugindata (Default to "Array ()")

plugin data

General structure

```
Default to "Array
(
)
" //plugin data
object {
assignfeedbackcomments_editor Optional //Editor structure
object {
text string //The text for this feedback.
format int //The format for this feedback
}
files_filemanager int Optional //The id of a draft area containing files for this feedback.
}
```

**XML-RPC (PHP structure)**

```
[plugindata] =>
Array
(
[assignfeedbackcomments_editor] =>
Array
(
[text] => string
[format] => int
)
[files_filemanager] => int
)
```

REST (POST parameters)

```
plugindata[assignfeedbackcomments_editor][text]= string
plugindata[assignfeedbackcomments_editor][format]= int
plugindata[files_filemanager]= int
```

advancedgradingdata (Default to "Array ()")

advanced grading data

```
General structure

Default to "Array
(
)
" //advanced grading data
object {
guide Optional //items
object {
criteria list of (
object {
criterionid int //criterion id
fillings Optional //filling
list of (
object {
criterionid int //criterion id
levelid int Optional //level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
score double //maximum score
}
)
)
)
rubric Optional //items
object {
criteria list of (
object {
criterionid int //criterion id
fillings Optional //filling
list of (
object {
criterionid int //criterion id
levelid int Optional //level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
}
)
)
)
}
```



XML-RPC (PHP structure)

8

REST (POST parameters)

```
advancedgradingdata[guide][criteria][0][criterionid]= int  
advancedgradingdata[guide][criteria][0][fillings][0][criterionid]= int  
advancedgradingdata[guide][criteria][0][fillings][0][levelid]= int  
advancedgradingdata[guide][criteria][0][fillings][0][remark]= string  
advancedgradingdata[guide][criteria][0][fillings][0][remarkformat]= int  
advancedgradingdata[guide][criteria][0][fillings][0][score]= double  
advancedgradingdata[rubric][criteria][0][criterionid]= int  
advancedgradingdata[rubric][criteria][0][fillings][0][criterionid]= int  
advancedgradingdata[rubric][criteria][0][fillings][0][levelid]= int  
advancedgradingdata[rubric][criteria][0][fillings][0][remark]= string  
advancedgradingdata[rubric][criteria][0][fillings][0][remarkformat]= int
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_save_grades

Save multiple grade updates for an assignment.

Arguments**assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

applytoall (Required)

If true, this grade will be applied to all members of the group (for group assignments).

General structure

```
int //If true, this grade will be applied to all members of the group (for group assignments).
```

XML-RPC (PHP structure)

```
[applytoall] => int
```

REST (POST parameters)

```
applytoall= int
```

grades (Required)

```
General structure

list of (
object {
userid int //The student id to operate on
grade double //The new grade for this user. Ignored if advanced grading used
attemptnumber int //The attempt number (-1 means latest attempt)
addattempt int //Allow another attempt if manual attempt reopen method
workflowstate string //The next marking workflow state
plugindata Default to "Array
(
)
" //plugin data
object {
assignfeedbackcomments_editor Optional //Editor structure
object {
text string //The text for this feedback.
format int //The format for this feedback
}
files_filemanager int Optional //The id of a draft area containing files for this feedback.
}
advancedgradingdata Default to "Array
(
)
" //advanced grading data
object {
guide Optional //items
object {
criteria list of (
object {
criterionid int //criterion id
fillings Optional //filling
list of (
object {
criterionid int //criterion id
levelid int Optional //Level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
score double //maximum score
})
})
})
rubric Optional //items
object {
criteria list of (
object {
criterionid int //criterion id
fillings Optional //filling
list of (
object {
criterionid int //criterion id
levelid int Optional //Level id
remark string Optional //remark
remarkformat int Optional //remark format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
})
})
})
```



REST (POST parameters)

```

grades[0][userid]= int
grades[0][grade]= double
grades[0][attemptnumber]= int
grades[0][addattempt]= int
grades[0][workflowstate]= string
grades[0][plugindata][assignfeedbackcomments_editor][text]= string
grades[0][plugindata][assignfeedbackcomments_editor][format]= int
grades[0][plugindata][files_filemanager]= int
grades[0][advancedgradingdata][guide][criteria][0][criterionid]= int
grades[0][advancedgradingdata][guide][criteria][0][fillings][0][criterionid]= int
grades[0][advancedgradingdata][guide][criteria][0][fillings][0][levelid]= int
grades[0][advancedgradingdata][guide][criteria][0][fillings][0][remark]= string
grades[0][advancedgradingdata][guide][criteria][0][fillings][0][remarkformat]= int
grades[0][advancedgradingdata][guide][criteria][0][fillings][0][score]= double
grades[0][advancedgradingdata][rubric][criteria][0][criterionid]= int
grades[0][advancedgradingdata][rubric][criteria][0][fillings][0][criterionid]= int
grades[0][advancedgradingdata][rubric][criteria][0][fillings][0][levelid]= int
grades[0][advancedgradingdata][rubric][criteria][0][fillings][0][remark]= string
grades[0][advancedgradingdata][rubric][criteria][0][fillings][0][remarkformat]= int

```

Response**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_save_submission

Update the current students submission

Arguments**assignmentid (Required)**

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

plugindata (Required)

General structure

```
object {
    onlinetext_editor  Optional //Editor structure
    object {
        text string   //The text for this submission.
        format int    //The format for this submission
        itemid int    //The draft area id for files attached to the submission
    }
    files_filemanager int Optional //The id of a draft area containing files for this submission.
}
```

XML-RPC (PHP structure)

```
[plugindata] =>
    Array
        (
            [onlinetext_editor] =>
                Array
                    (
                        [text] => string
                        [format] => int
                        [itemid] => int
                    )
            [files_filemanager] => int
        )
```

REST (POST parameters)

```
plugindata[onlinetext_editor][text]= string
plugindata[onlinetext_editor][format]= int
plugindata[onlinetext_editor][itemid]= int
plugindata[files_filemanager]= int
```

Response

list of warnings

General structure

```
Optional //list of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

mod_assign_save_user_extensions

Save a list of assignment extensions

Arguments**assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userids (Required)

1 or more user ids

General structure

```
//1 or more user ids
list of (
int //user id
)
```

XML-RPC (PHP structure)

```
[userids] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
userids[0]= int
```

dates (Required)

1 or more extension dates (timestamp)

General structure

```
//1 or more extension dates (timestamp)
list of (
int //dates
)
```

**XML-RPC (PHP structure)**

```
[dates] =>
Array
(
[0] => int
)
```

REST (POST parameters)

```
dates[0]= int
```

Response

list of warnings

General structure

```

Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)

```

XML-RPC (PHP structure)

```

Array
(
    [0] =>
        Array
        (
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_set_user_flags

Creates or updates user flags

Arguments**assignmentid** (Required)

assignment id

General structure

```
int //assignment id
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userflags (Required)

General structure

```
list of (
object {
userid int //student id
locked int Optional //Locked
mailed int Optional //mailed
extensionduedate int Optional //extension due date
workflowstate string Optional //marking workflow state
allocatedmarker int Optional //allocated marker
}
)
```

XML-RPC (PHP structure)

```
[userflags] =>
  Array
    (
      [0] =>
        Array
          (
            [userid] => int
            [locked] => int
            [mailed] => int
            [extensionduedate] => int
            [workflowstate] => string
            [allocatedmarker] => int
          )
    )
  )
```

**REST (POST parameters)**

```
userflags[0][userid]= int
userflags[0][locked]= int
userflags[0][mailed]= int
userflags[0][extensionduedate]= int
userflags[0][workflowstate]= string
userflags[0][allocatedmarker]= int
```

Response

General structure

```
list of (
object {
id int //id of record if successful, -1 for failure
userid int //userid of record
errormessage string Optional //Failure error message
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[id] => int
[userid] => int
[errormessage] => string
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="errormessage">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
<MESSAGE>Invalid parameter value detected</MESSAGE>
<DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

mod_assign_start_submission

Start a submission for user if assignment has a time limit.

Arguments

assignnid (Required)

Assignment instance id

General structure

```
int //Assignment instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

REST (POST parameters)

```
assignid= int
```

Response**General structure**

```
object {
    submissionid int //New submission ID.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [submissionid] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="submissionid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_assign_submit_for_grading**

Submit the current students assignment for grading

Arguments**assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

acceptsubmissionstatement (Required)

Accept the assignment submission statement

General structure

```
int //Accept the assignment submission statement
```

XML-RPC (PHP structure)

```
[acceptsubmissionstatement] => int
```

REST (POST parameters)

```
acceptsubmissionstatement= int
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

XML-RPC (PHP structure)

```
Array
(
    [0] =>
        Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_submit_grading_form

Submit the grading form data via ajax

**Arguments****assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userid (Required)

The user id the submission belongs to

General structure

```
int //The user id the submission belongs to
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

jsonformdata (Required)

The data from the grading form, encoded as a json array

General structure

```
string //The data from the grading form, encoded as a json array
```

XML-RPC (PHP structure)

```
[jsonformdata] => string
```

REST (POST parameters)

```
jsonformdata= string
```

Response

list of warnings

General structure

```
Optional //List of warnings
list of (
    //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
```

**XML-RPC (PHP structure)**

```
Array
(
    [0] =>
        Array
        (
            [item] => string
            [itemid] => int
            [warningcode] => string
            [message] => string
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_assign_unlock_submissions

Allow students to make changes to a list of submissions

**Arguments****assignmentid** (Required)

The assignment id to operate on

General structure

```
int //The assignment id to operate on
```

XML-RPC (PHP structure)

```
[assignmentid] => int
```

REST (POST parameters)

```
assignmentid= int
```

userids (Required)

1 or more user ids

General structure

```
//1 or more user ids
list of (
int //user id
)
```

XML-RPC (PHP structure)

```
[userids] =>
Array
(
    [0] => int
)
```

REST (POST parameters)

```
userids[0]= int
```

Response

list of warnings

General structure

```
Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```

XML-RPC (PHP structure)

```
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_view_assign

Update the module completion status.

**Arguments**

assignid (Required)

assign instance id

General structure

```
int //assign instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

REST (POST parameters)

```
assignid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_view_grading_table

Trigger the grading_table_viewed event.

Arguments**assignid** (Required)

assign instance id

General structure

```
int //assign instance id
```

XML-RPC (PHP structure)

```
[assignid] => int
```

REST (POST parameters)

```
assignid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )}
```

XML-RPC (PHP structure)

```
Array
  (
    [status] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_assign_view_submission_status

Trigger the submission status viewed event.

Arguments

`assignnid` (Required)

assign instance id

General structure

```
int //assign instance id
```

XML-RPC (PHP structure)

```
[assignnid] => int
```

REST (POST parameters)

```
assignnid= int
```

Response

General structure

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_bigbluebuttonbn_can_join**

Returns information if the current user can join or not.

Arguments**cmid** (Required)

course module id

General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

groupid (Default to "0")

bigbluebuttonbn group id

General structure

```
int Default to "0" //bigbluebuttonbn group id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
    can_join int //Can join session
    cmid int //course module id
}
```

XML-RPC (PHP structure)

```
Array
(
    [can_join] => int
    [cmid] => int
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="can_join">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cmid">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_bigbluebuttonbn_completion_validate

Validate completion

Arguments

bigbluebuttonbnid (Required)

bigbluebuttonbn instance id

General structure

```
int //bigbluebuttonbn instance id
```

XML-RPC (PHP structure)

```
[bigbluebuttonbnid] => int
```

REST (POST parameters)

```
bigbluebuttonbnid= int
```

Response

General structure

```
object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```



XML-RPC (PHP structure)

```
Array
(
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_end_meeting

End a meeting

Arguments

bigbluebuttonbnid (Required)

bigbluebuttonbn instance id

General structure

int //bigbluebuttonbn instance id

XML-RPC (PHP structure)

[bigbluebuttonbnid] => int

REST (POST parameters)

```
bigbluebuttonbnid= int
```

groupid (Default to "0")

bigbluebuttonbn group id

General structure

```
int Default to "0" //bigbluebuttonbn group id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_get_bigbluebuttonbns_by_courses

Returns a list of bigbluebuttonbns in a provided list of courses, if no list is provided all bigbluebuttonbns that the user can view will be returned.

Arguments

courseids (Default to "Array ()")

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //Course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
courseids[0]= int
```

Response**General structure**

```
object {
    bigbluebuttonbns list of (
        object {
            id int   //Activity instance id
            coursemodule int //Course module id
            course int //Course id
            name string //Activity name
            intro string //Activity introduction
            introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
            introfiles Optional //Files in the introduction
            list of (
                //File
            object {
                filename string Optional //File name.
                filepath string Optional //File path.
                filesize int Optional //File size.
                fileurl string Optional //Downloadable file url.
                timemodified int Optional //Time modified.
                mimetype string Optional //File mime type.
                isexternalfile int Optional //Whether is an external file.
                repositorytype string Optional //The repository type for external files.
            }
        )
        section int Optional //Course section id
        visible int Optional //Visible
        groupmode int Optional //Group mode
        groupingid int Optional //Group id
        lang string Optional //Forced activity language
        meetingid string //Meeting id
        timemodified int //Last time the instance was modified
    )
    )warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
})
```



XML-RPC (PHP structure)

```
Array
(
    [bigbluebuttonbns] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [coursemodule] => int
                            [course] => int
                            [name] => string
                            [intro] => string
                            [introformat] => int
                            [introfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [section] => int
                            [visible] => int
                            [groupmode] => int
                            [groupingid] => int
                            [lang] => string
                            [meetingid] => string
                            [timemodified] => int
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)
```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="bigbluebuttonbns">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursemodule">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="course">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="intro">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="introformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="introfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="repositorytype">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="section">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupmode">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupingid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="lang">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="meetingid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>

```

```

<KEY name="item">
    <VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_bigbluebuttonbn_get_join_url

Get the join URL for the meeting and create if it does not exist.

Arguments

cmid (Required)

course module id



General structure

```
int //course module id
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

groupid (Default to "0")

bigbluebuttonbn group id

General structure

```
int Default to "0" //bigbluebuttonbn group id
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
join_url string Optional //Can join session
warnings Optional //List of warnings
list of (
 /warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
[join_url] => string
[warnings] =>
    Array
    (
[0] =>
        Array
        (
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="join_url">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_bigbluebuttonbn_get_recordings**

Returns a list of recordings ready to be processed by a datatable.

Arguments

bigbluebuttonbnid (Required)

bigbluebuttonbn instance id

General structure

```
int //bigbluebuttonbn instance id
```

XML-RPC (PHP structure)

```
[bigbluebuttonbnid] => int
```

REST (POST parameters)

```
bigbluebuttonnid= int
```

tools (Default to "protect,unprotect,publish,unpublish,delete")

a set of enabled tools

General structure

```
string Default to "protect,unprotect,publish,unpublish,delete" //a set of enabled tools
```

XML-RPC (PHP structure)

```
[tools] => string
```

REST (POST parameters)

```
tools= string
```

groupid (Default to "null")

Group ID

General structure

```
int Default to "null" //Group ID
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
status int //Whether the fetch was successful
tabledata object {
activity string ping_interval int locale string profile_features list of (
string )columns list of (
object {
key string label string width string type string Optional //Column type
sortable int Optional //Whether this column is sortable
allowHTML int Optional //Whether this column contains HTML
formatter string Optional //Formatter name
})
)data string }
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [tabledata] =>
        Array
            (
                [activity] => string
                [ping_interval] => int
                [locale] => string
                [profile_features] =>
                    Array
                        (
                            [0] => string
                        )
                [columns] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [key] => string
                                        [label] => string
                                        [width] => string
                                        [type] => string
                                        [sortable] => int
                                        [allowHTML] => int
                                        [formatter] => string
                                    )
                        )
                [data] => string
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="tabledata">
            <SINGLE>
                <KEY name="activity">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="ping_interval">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="locale">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="profile_features">
                    <MULTIPLE>
                        <VALUE>string</VALUE>
                    </MULTIPLE>
                </KEY>
                <KEY name="columns">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="key">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="label">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="width">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="type">
                                <VALUE>string</VALUE>
                            </KEY>
                            <KEY name="sortable">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="allowHTML">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="formatter">
                                <VALUE>string</VALUE>
                            </KEY>
                        </SINGLE>
                    </MULTIPLE>
                </KEY>
                <KEY name="data">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_get_recordings_to_import

Returns a list of recordings ready to import to be processed by a datatable.

Arguments**destinationinstanceid** (Required)

Id of the other BBB we target for importing recordings into. The idea here is to remove already imported recordings.

General structure

```
int //Id of the other BBB we target for importing recordings into.
    The idea here is to remove already imported recordings.
```

XML-RPC (PHP structure)

```
[destinationinstanceid] => int
```

REST (POST parameters)

```
destinationinstanceid= int
```

sourcebigbluebuttonbnid (Default to "0")

bigbluebuttonbn instance id

**General structure**

```
int Default to "0" //bigbluebuttonbn instance id
```

XML-RPC (PHP structure)

```
[sourcebigbluebuttonbnid] => int
```

REST (POST parameters)

```
sourcebigbluebuttonbnid= int
```

sourcecourseid (Default to "0")

source courseid to filter by

General structure

```
int Default to "0" //source courseid to filter by
```

XML-RPC (PHP structure)

```
[sourcecourseid] => int
```

REST (POST parameters)

```
sourcecourseid= int
```

tools (Default to "protect,unprotect,publish,unpublish,delete")

a set of enabled tools

General structure

```
string Default to "protect,unprotect,publish,unpublish,delete" //a set of enabled tools
```

XML-RPC (PHP structure)

```
[tools] => string
```

REST (POST parameters)

```
tools= string
```

groupid (Default to "null")

Group ID

General structure

```
int Default to "null" //Group ID
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
status int //Whether the fetch was successful
tabledata object {
activity string ping_interval int locale string profile_features list of (
string )columns list of (
object {
key string label string width string type string Optional //Column type
sortable int Optional //Whether this column is sortable
allowHTML int Optional //Whether this column contains HTML
})
)data string }
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [tabledata] =>
        Array
            (
                [activity] => string
                [ping_interval] => int
                [locale] => string
                [profile_features] =>
                    Array
                        (
                            [0] => string
                        )
                [columns] =>
                    Array
                        (
                            [0] =>
                                Array
                                    (
                                        [key] => string
                                        [label] => string
                                        [width] => string
                                        [type] => string
                                        [sortable] => int
                                        [allowHTML] => int
                                    )
                            )
                [data] => string
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="status">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="tabledata">
      <SINGLE>
        <KEY name="activity">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="ping_interval">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="locale">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="profile_features">
          <MULTIPLE>
            <VALUE>string</VALUE>
          </MULTIPLE>
        </KEY>
        <KEY name="columns">
          <MULTIPLE>
            <SINGLE>
              <KEY name="key">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="label">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="width">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="type">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="sortable">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="allowHTML">
                <VALUE>int</VALUE>
              </KEY>
            </SINGLE>
          </MULTIPLE>
        </KEY>
        <KEY name="data">
          <VALUE>string</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_meeting_info

Get displayable information on the meeting

Arguments**bigbluebuttonnid** (Required)

bigbluebuttonbn instance id

General structure

```
int //bigbluebuttonbn instance id
```

XML-RPC (PHP structure)

```
[bigbluebuttonnid] => int
```

REST (POST parameters)

```
bigbluebuttonnid= int
```

groupid (Default to "0")

bigbluebuttonbn group id

General structure

```
int Default to "0" //bigbluebuttonbn group id
```

**XML-RPC (PHP structure)**

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

updatecache (Default to "")

update cache ?

General structure

```
int Default to "" //update cache ?
```

XML-RPC (PHP structure)

[updatecache] => int

REST (POST parameters)

updatecache= int

Response**General structure**

```
object {
    cmid int //CM id
    userlimit int //User limit
    bigbluebuttonnid string //bigbluebuttonbn instance id
    groupid int Default to "0" //bigbluebuttonbn group id
    meetingid string //Meeting id
    openingtime int Optional //Opening time
    closingtime int Optional //Closing time
    statusrunning int Optional //Status running
    statusclosed int Optional //Status closed
    statusopen int Optional //Status open
    statusmessage string Optional //Status message
    startedat int Optional //Started at
    moderatorcount int Optional //Moderator count
    participantcount int Optional //Participant count
    moderatorplural int Optional //Several moderators ?
    participantplural int Optional //Several participants ?
    canjoin int //Can join
    ismoderator int //Is moderator
    presentations list of (
        object {
            url string //presentation URL
            iconname string //icon name
            icondesc string //icon text
            name string //presentation name
        }
    )joinurl string //Join URL
    guestaccesenabled int Optional //Guest access enabled
    guestjoinurl string Optional //Guest URL
    guestpassword string Optional //Guest join password
    features Optional //List of features for the instance
    list of (
        object {
            name string //Feature name.
            isenabled int //Whether the feature is enabled.
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [cmid] => int
    [userlimit] => int
    [bigbluebuttonnid] => string
    [groupid] => int
    [meetingid] => string
    [openingtime] => int
    [closingtime] => int
    [statusrunning] => int
    [statusclosed] => int
    [statusopen] => int
    [statusmessage] => string
    [startedat] => int
    [moderatorcount] => int
    [participantcount] => int
    [moderatorplural] => int
    [participantplural] => int
    [canjoin] => int
    [ismoderator] => int
    [presentations] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [url] => string
                            [iconname] => string
                            [icondesc] => string
                            [name] => string
                        )
            )
    [joinurl] => string
    [guestaccessenabled] => int
    [guestjoinurl] => string
    [guestpassword] => string
    [features] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [name] => string
                            [isEnabled] => int
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="cmdid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="userlimit">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="bigbluebuttonnid">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="groupid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="meetingid">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="openingtime">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="closingtime">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusrunning">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusclosed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusopen">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="statusmessage">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="startedat">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="moderatorcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="participantcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="moderatorplural">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="participantplural">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canjoin">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="ismoderator">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="presentations">
      <MULTIPLE>
        <SINGLE>
          <KEY name="url">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="iconname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="icondesc">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="joinurl">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="guestaccesenabled">
      <VALUE>int</VALUE>
    </KEY>
```



```

<KEY name="guestjoinurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="guestpassword">
    <VALUE>string</VALUE>
</KEY>
<KEY name="features">
    <MULTIPLE>
        <SINGLE>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isEnabled">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_update_recording

Update a single recording



Arguments

bigbluebuttonbnid (Required)

bigbluebuttonbn instance id, this might be a different one from the one set in recordingid in case of importing

General structure

```
int //bigbluebuttonbn instance id, this might be a different one from the one set in recordingid in case of importing
```

XML-RPC (PHP structure)

```
[bigbluebuttonbnid] => int
```

REST (POST parameters)

```
bigbluebuttonbnid= int
```

recordingid (Required)

The moodle internal recording ID

General structure

```
int //The moodle internal recording ID
```

XML-RPC (PHP structure)

```
[recordingid] => int
```

REST (POST parameters)

```
recordingid= int
```

action (Required)

The action to perform

General structure

```
string //The action to perform
```

XML-RPC (PHP structure)

```
[action] => string
```

REST (POST parameters)

```
action= string
```

additionaloptions (Required)

Additional options

General structure

```
string //Additional options
```

XML-RPC (PHP structure)

```
[additionaloptions] => string
```

REST (POST parameters)

```
additionaloptions= string
```

**Response****General structure**

```
object {  
}
```

XML-RPC (PHP structure)

```
Array  
(  
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_bigbluebuttonbn_view_bigbluebuttonbn

Trigger the course module viewed event and update the module completion status.

Arguments**bigbluebuttonbnid** (Required)

bigbluebuttonbn instance id

General structure

```
int //bigbluebuttonbn instance id
```

XML-RPC (PHP structure)

```
[bigbluebuttonbnid] => int
```

REST (POST parameters)

```
bigbluebuttonbnid= int
```

Response

General structure

```

object {
    status int   //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string  Optional //item
        itemid int  Optional //item id
        warningcode string  //the warning code can be used by the client app to implement specific behaviour
        message string  //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_book_get_books_by_courses

Returns a list of book instances in a provided set of courses, if no courses are provided then all the book instances the user has access to will be returned.

Arguments

courseids (Default to "Array ()")

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
)
```

REST (POST parameters)

```
courseids[0]= int
```

Response



General structure

```
object {
books list of (
    //Books
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
    //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
numbering int //Book numbering configuration
navstyle int //Book navigation style configuration
customtitles int //Book custom titles type
revision int Optional //Book revision
timecreated int Optional //Time of creation
timemodified int Optional //Time of last modification
}
)warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```



XML-RPC (PHP structure)

```
Array
(
    [books] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [coursemodule] => int
                            [course] => int
                            [name] => string
                            [intro] => string
                            [introformat] => int
                            [introfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [section] => int
                            [visible] => int
                            [groupmode] => int
                            [groupingid] => int
                            [lang] => string
                            [numbering] => int
                            [navstyle] => int
                            [customtitles] => int
                            [revision] => int
                            [timecreated] => int
                            [timemodified] => int
                        )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="books">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursemodule">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="course">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="intro">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="introformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="introfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="repositorytype">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="section">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupmode">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupingid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="lang">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="numbering">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="navstyle">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="customtitles">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="revision">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```

<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No



mod_book_view_book

Simulate the view.php web interface book: trigger events, completion, etc...

Arguments

bookid (Required)

book instance id

General structure

```
int //book instance id
```

XML-RPC (PHP structure)

```
[bookid] => int
```

REST (POST parameters)

bookid= int

chapterid (Default to "0")

chapterid

General structureint **Default to "0"** //chapter id**XML-RPC (PHP structure)**

[chapterid] => int

REST (POST parameters)

chapterid= int

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_chat_get_chat_latest_messages**

Get the latest messages from the given chat session.

Arguments**chatsid** (Required)

chat session id (obtained via mod_chat_login_user)

General structure

```
string //chat session id (obtained via mod_chat_login_user)
```

XML-RPC (PHP structure)

```
[chatsid] => string
```

REST (POST parameters)

chatsid= string

chatlasttime (Default to "0")

last time messages were retrieved (epoch time)

General structure

int Default to "0" //Last time messages were retrieved (epoch time)

XML-RPC (PHP structure)

[chatlasttime] => int

REST (POST parameters)

chatlasttime= int

Response**General structure**

```
object {
  messages //List of users
  list of (
    object {
      id int //message id
      userid int //user id
      system int //true if is a system message (like user joined)
      message string //message text
      timestamp int //timestamp for the message
    }
    chatnewlasttime int //new last time
    warnings Optional //List of warnings
    list of (
      //warning
      object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
      }
    )
  }
}
```

XML-RPC (PHP structure)

```
Array
(
    [messages] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [userid] => int
                            [system] => int
                            [message] => string
                            [timestamp] => int
                        )
                )
            )
    [chatnewlasttime] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="messages">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="system">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestamp">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="chatnewlasttime">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_chat_get_chat_users

Get the list of users in the given chat session.

Arguments

chatsid (Required)

chat session id (obtained via mod_chat_login_user)

General structure

```
string //chat session id (obtained via mod_chat_login_user)
```

XML-RPC (PHP structure)

```
[chatsid] => string
```

REST (POST parameters)

```
chatsid= string
```

Response

General structure

```
object {
  users //List of users
  list of (
    object {
      id int //user id
      fullname string //user full name
      profileimageurl string //user picture URL
    }
  )warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
  [users] =>
  Array
  (
    [0] =>
    Array
    (
      [id] => int
      [fullname] => string
      [profileimageurl] => string
    )
  )
  [warnings] =>
  Array
  (
    [0] =>
    Array
    (
      [item] => string
      [itemid] => int
      [warningcode] => string
      [message] => string
    )
  )
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="profileimageurl">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_chat_get_chats_by_courses

Returns a list of chat instances in a provided set of courses, if no courses are provided then all the chat instances the user has access to will be returned.

Arguments**courseids** (Default to "Array ()")

Array of course ids

```
General structure

Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

```
XML-RPC (PHP structure)

[courseids] =>
    Array
        (
            [0] => int
        )
```

```
REST (POST parameters)

courseids[0]= int
```

Response

```
General structure

object {
chats list of (
    //Chats
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
    //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
chatmethod string Optional //chat method (sockets, ajax, header_js)
keepdays int Optional //keep days
studentlogs int Optional //student logs visible to everyone
chattime int Optional //chat time
schedule int Optional //schedule type
timemodified int Optional //time of last modification
}
>warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [chats] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [coursemodule] => int
                            [course] => int
                            [name] => string
                            [intro] => string
                            [introformat] => int
                            [introfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [section] => int
                            [visible] => int
                            [groupmode] => int
                            [groupingid] => int
                            [lang] => string
                            [chatmethod] => string
                            [keepdays] => int
                            [studentlogs] => int
                            [chattime] => int
                            [schedule] => int
                            [timemodified] => int
                        )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="chats">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="coursemodule">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="course">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="intro">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="introformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="introfiles">
            <MULTIPLE>
              <SINGLE>
                <KEY name="filename">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filepath">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filesize">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="fileurl">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="timemodified">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="mimetype">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="isexternalfile">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="repositorytype">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="section">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="visible">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="groupmode">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="groupingid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="lang">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="chatmethod">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="keepdays">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="studentlogs">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="chattime">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```

<KEY name="schedule">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No



mod_chat_get_session_messages

Retrieves messages of the given chat session.

Arguments

chatid (Required)

Chat instance id.

General structure

```
int //Chat instance id.
```

XML-RPC (PHP structure)

```
[chatid] => int
```

REST (POST parameters)

chatid= int

sessionstart (Required)

The session start time (timestamp).

General structure

```
int //The session start time (timestamp).
```

XML-RPC (PHP structure)

```
[sessionstart] => int
```

REST (POST parameters)

sessionstart= int

sessionend (Required)

The session end time (timestamp).

General structure

```
int //The session end time (timestamp).
```

XML-RPC (PHP structure)

```
[sessionend] => int
```

REST (POST parameters)

sessionend= int

groupid (Default to "0")

Get messages from users in this group. 0 means that the function will determine the user group

General structure

```
int Default to "0" //Get messages from users in this group.  
0 means that the function will determine the user group
```

**XML-RPC (PHP structure)**

```
[groupid] => int
```

REST (POST parameters)

groupid= int

Response

General structure

```

object {
  messages list of (
    object {
      id int   //The message record id.
      chatid int //The chat id.
      userid int //The user who wrote the message.
      groupid int //The group this message belongs to.
      issystem int //Whether is a system message or not.
      message string //The message text.
      timestamp int //The message timestamp (indicates when the message was sent).
    }
  )warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}

```

XML-RPC (PHP structure)

```

Array
(
  [messages] =>
    Array
      (
        [0] =>
          Array
            (
              [id] => int
              [chatid] => int
              [userid] => int
              [groupid] => int
              [issystem] => int
              [message] => string
              [timestamp] => int
            )
      )
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="messages">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="chatid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="issystem">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timestamp">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_chat_get_sessions

Retrieves chat sessions for a given chat.

Arguments

chatid (Required)

Chat instance id.

General structure

```
int //Chat instance id.
```

XML-RPC (PHP structure)

```
[chatid] => int
```

REST (POST parameters)

```
chatid= int
```

groupid (Default to "0")

Get messages from users in this group. 0 means that the function will determine the user group

General structure

```
int Default to "0" //Get messages from users in this group.  
0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

showall (Default to "")

Whether to show completed sessions or not.

General structure

```
int Default to "" //Whether to show completed sessions or not.
```

XML-RPC (PHP structure)

```
[showall] => int
```

REST (POST parameters)

```
showall= int
```

Response

General structure

```
object {
sessions  //list of users
list of (
object {
sessionstart int  //Session start time.
sessionend int  //Session end time.
sessionusers  //Session users.
list of (
object {
userid int  //User id.
messagecount int  //Number of messages in the session.
}
)iscomplete int  //Whether the session is completed or not.
}
)warnings  Optional //List of warnings
list of (
 //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)
})}
```

XML-RPC (PHP structure)

```
Array
(
    [sessions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [sessionstart] => int
                            [sessionend] => int
                            [sessionusers] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [userid] => int
                                                    [messagecount] => int
                                                )
                                    )
                            [iscomplete] => int
                        )
                )
)
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="sessions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="sessionstart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="sessionend">
                        <VALUE>int</VALUE>
                    </KEY>
                <KEY name="sessionusers">
                    <MULTIPLE>
                        <SINGLE>
                            <KEY name="userid">
                                <VALUE>int</VALUE>
                            </KEY>
                            <KEY name="messagecount">
                                <VALUE>int</VALUE>
                            </KEY>
                        </SINGLE>
                    </MULTIPLE>
                </KEY>
                <KEY name="iscomplete">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <SINGLE>
                <KEY name="item">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="itemid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="warningcode">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="message">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <SINGLE>
        </RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_chat_login_user

Log a user into a chat room in the given chat.

Arguments**chatid** (Required)

chat instance id

General structure

```
int //chat instance id
```

XML-RPC (PHP structure)

```
[chatid] => int
```

REST (POST parameters)

```
chatid= int
```

groupid (Default to "0")

group id, 0 means that the function will determine the user group

General structure

```
int Default to "0" //group id, 0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response**General structure**

```
object {
  chatsid string //unique chat session id
  warnings Optional //list of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
  (
    [chatsid] => string
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="chatsid">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_chat_send_chat_message`

Send a message on the given chat session.

Arguments

chatsid (Required)

chat session id (obtained via mod_chat_login_user)

General structure

```
string //chat session id (obtained via mod_chat_login_user)
```

XML-RPC (PHP structure)

```
[chatsid] => string
```

REST (POST parameters)

```
chatsid= string
```

messagetext (Required)

the message text

General structure

```
string //the message text
```

XML-RPC (PHP structure)

```
[messagetext] => string
```

REST (POST parameters)

```
messagetext= string
```

beepid (Default to "")

the beep id

General structure

```
string Default to "" //the beep id
```

XML-RPC (PHP structure)

```
[beepid] => string
```



REST (POST parameters)

```
beepid= string
```

Response

General structure

```

object {
  messageid int    //message sent id
  warnings  Optional //list of warnings
  list of (
    //warning
  object {
    item string  Optional //item
    itemid int   Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}

```

XML-RPC (PHP structure)

```

Array
(
  [messageid] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="messageid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_chat_view_chat`

Trigger the course module viewed event and update the module completion status.

Arguments

`chatid` (Required)

chat instance id

General structure

```
int //chat instance id
```

XML-RPC (PHP structure)

```
[chatid] => int
```

REST (POST parameters)

```
chatid= int
```

Response

General structure

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```



XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_choice_delete_choice_responses**

Delete the given submitted responses in a choice

Arguments

choiceid (Required)

choice instance id

General structure

int //choice instance id

XML-RPC (PHP structure)

[choiceid] => int

REST (POST parameters)

```
choiceid= int
```

responses (Default to "Array ()")

Array of response ids, empty for deleting all the current user responses.

General structure

```
Default to "Array
(
)
" //Array of response ids, empty for deleting all the current user responses.
list of (
int  //response id
)
```

XML-RPC (PHP structure)

```
[responses] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
responses[0]= int
```

Response**General structure**

```
object {
status int  //status, true if everything went right
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_choice_get_choice_options**

Retrieve options for a specific choice.

Arguments

choiceid (Required)

choice instance id

General structure

```
int //choice instance id
```

XML-RPC (PHP structure)

```
[choiceid] => int
```

REST (POST parameters)

```
choiceid= int
```

Response

```
General structure

object {
    options //Options
    list of (
        object {
            id int //option id
            text string //text of the choice
            maxanswers int //maximum number of answers
            displaylayout int //true for orizontal, otherwise vertical
            countanswers int //number of answers
            checked int //we already answered
            disabled int //option disabled
        }
    )warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```

XML-RPC (PHP structure)

```
Array
(
    [options] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [text] => string
                        [maxanswers] => int
                        [displaylayout] => int
                        [countanswers] => int
                        [checked] => int
                        [disabled] => int
                    )
                )
        )
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="options">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="text">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="maxanswers">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="displaylayout">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="countanswers">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="checked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="disabled">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_choice_get_choice_results

Retrieve users results for a given choice.

Arguments

choiceid (Required)

choice instance id

General structure

```
int //choice instance id
```

XML-RPC (PHP structure)

```
[choiceid] => int
```

REST (POST parameters)

```
choiceid= int
```

Response**General structure**

```
object {
  options list of (
    //Options
  object {
    id int //choice instance id
    text string //text of the choice
    maxanswer int //maximum number of answers
    userresponses list of (
      //User responses
      object {
        userid int //user id
        fullname string //user full name
        profileimageurl string //profile user image url
        answerid int Optional //answer id
        timemodified int Optional //time of modification
      }
    )numberofuser int //number of users answers
    percentageamount double //percentage of users answers
  }
  )warnings Optional //list of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```



XML-RPC (PHP structure)

```
Array
(
    [options] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [text] => string
                            [maxanswer] => int
                            [userresponses] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [userid] => int
                                                    [fullname] => string
                                                    [profileimageurl] => string
                                                    [answerid] => int
                                                    [timemodified] => int
                                                )
                                        )
                            [numberofuser] => int
                            [percentageamount] => double
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="options">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="text">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="maxanswer">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userresponses">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fullname">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="profileimageurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="answerid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                    <KEY name="numberofuser">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="percentageamount">
                        <VALUE>double</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_choice_get_choices_by_courses

Returns a list of choice instances in a provided set of courses, if no courses are provided then all the choice instances the user has access to will be returned.

Arguments**courseids** (Default to "Array ()")

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

courseids[0]= int

**Response**

General structure

```
object {
choices list of (
 //Choices
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
publish int Optional //If choice is published
showresults int Optional //0 never, 1 after answer, 2 after close, 3 always
display int Optional //Display mode (vertical, horizontal)
allowupdate int Optional //Allow update
allowmultiple int Optional //Allow multiple choices
showunanswered int Optional //Show users who not answered yet
includeinactive int Optional //Include inactive users
limitanswers int Optional //Limit answers
timeopen int Optional //Date of opening validity
timeclose int Optional //Date of closing validity
showpreview int Optional //Show preview before timeopen
timemodified int Optional //Time of last modification
completionsubmit int Optional //Completion on user submission
showavailable int Optional //Show available spaces
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

XML-RPC (PHP structure)

```
Array
(
    [choices] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [coursemodule] => int
                            [course] => int
                            [name] => string
                            [intro] => string
                            [introformat] => int
                            [introfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [section] => int
                            [visible] => int
                            [groupmode] => int
                            [groupingid] => int
                            [lang] => string
                            [publish] => int
                            [showresults] => int
                            [display] => int
                            [allowupdate] => int
                            [allowmultiple] => int
                            [showunanswered] => int
                            [includeinactive] => int
                            [limitanswers] => int
                            [timeopen] => int
                            [timeclose] => int
                            [showpreview] => int
                            [timemodified] => int
                            [completionsubmit] => int
                            [showavailable] => int
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="choices">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursemodule">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="course">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="intro">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="introformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="introfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="repositorytype">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="section">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupmode">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupingid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="lang">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="publish">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showresults">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="display">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="allowupdate">
            <VALUE>int</VALUE>
        </KEY>
```



```

<KEY name="allowmultiple">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showunanswered">
    <VALUE>int</VALUE>
</KEY>
<KEY name="includeinactive">
    <VALUE>int</VALUE>
</KEY>
<KEY name="limitanswers">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeopen">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timeclose">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showpreview">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completionsubmit">
    <VALUE>int</VALUE>
</KEY>
<KEY name="showavailable">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_choice_submit_choice_response

Submit responses to a specific choice item.

Arguments

choiceid (Required)

choice instance id

General structure

```
int //choice instance id
```

XML-RPC (PHP structure)

```
[choiceid] => int
```

REST (POST parameters)

```
choiceid= int
```

responses (Required)

Array of response ids

General structure

```
//Array of response ids
list of (
int //answer id
)
```

XML-RPC (PHP structure)

```
[responses] =>
  Array
    (
      [0] => int
    )
```

REST (POST parameters)

```
responses[0]= int
```

Response

General structure

```

object {
answers list of (
    //Answers
object {
id int //answer id
choiceid int //choiceid
userid int //user id
optionid int //optionid
timemodified int //time of last modification
}
)warnings Optional //list of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)

```

XML-RPC (PHP structure)

```

Array
(
[answers] =>
    Array
        (
[0] =>
        Array
            (
[id] => int
[choiceid] => int
[userid] => int
[optionid] => int
[timemodified] => int
)
)
[warnings] =>
    Array
        (
[0] =>
        Array
            (
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="answers">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="choiceid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="optionid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_choice_view_choice

Trigger the course module viewed event and update the module completion status.

Arguments

choiceid (Required)

choice instance id

General structure

```
int //choice instance id
```

XML-RPC (PHP structure)

```
[choiceid] => int
```

REST (POST parameters)

```
choiceid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_data_add_entry**

Adds a new entry.

Arguments**databaseid** (Required)

data instance id

General structure

int //data instance id

XML-RPC (PHP structure)

[databaseid] => int

REST (POST parameters)

```
databaseid= int
```

groupid (Default to "0")

Group id, 0 means that the function will determine the user group

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

data (Required)

The fields data to be created

General structure

```
//The fields data to be created
list of (
object {
fieldid int //The field id.
subfield string Default to "" //The subfield name (if required).
value string //The contents for the field always JSON encoded.
}
)
```

XML-RPC (PHP structure)

```
[data] =>
  Array
    (
      [0] =>
        Array
          (
            [
              [fieldid] => int
              [subfield] => string
              [value] => string
            )
      )
    )
```

**REST (POST parameters)**

```
data[0][fieldid]= int
data[0][subfield]= string
data[0][value]= string
```

Response

General structure

```
object {
newentryid int   //True new created entry id. 0 if the entry was not created.
generalnotifications list of (
string  //General notifications
)fieldnotifications list of (
object {
fieldname string  //The field name.
notification string  //The notification for the field.
}
)warnings  Optional //List of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)}
})}
```

XML-RPC (PHP structure)

```
    Array
      (
        [newentryid] => int
        [generalnotifications] =>
          Array
            (
              [
                [0] => string
              ]
            )
        [fieldnotifications] =>
          Array
            (
              [
                [0] =>
                  Array
                    (
                      [
                        [fieldname] => string
                        [notification] => string
                      ]
                    )
              ]
            )
      )
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  ]
                )
          ]
        )
    )
  )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="newentryid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="generalnotifications">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="fieldnotifications">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="fieldname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="notification">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_approve_entry

Approves or unapproves an entry.

Arguments

entryid (Required)

Record entry id.

General structure

```
int //Record entry id.
```

XML-RPC (PHP structure)

```
[entryid] => int
```

REST (POST parameters)

```
entryid= int
```

approve (Default to "1")

Whether to approve (true) or unapprove the entry.

General structure

```
int Default to "1" //Whether to approve (true) or unapprove the entry.
```

XML-RPC (PHP structure)

```
[approve] => int
```

REST (POST parameters)

```
approve= int
```

Response**General structure**

```
object {
  status int //status: true if success
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
  [status] => int
  [warnings] =>
    Array
    (
      [0] =>
        Array
        (
          [item] => string
          [itemid] => int
          [warningcode] => string
          [message] => string
        )
    )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

[mod_data_delete_entry](#)

Deletes an entry.

Arguments**entryid** (Required)

Record entry id.

General structure

int //Record entry id.

XML-RPC (PHP structure)

[entryid] => int

REST (POST parameters)

```
entryid= int
```

Response**General structure**

```
object {
    status int //Always true. If we see this field it means that the entry was deleted.
    warnings Optional //List of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_delete_saved_preset

Delete site user preset.

Arguments

dataid (Required)

Id of the data activity

General structure

```
int //Id of the data activity
```

XML-RPC (PHP structure)

```
[dataid] => int
```

REST (POST parameters)

```
dataid= int
```

presetnames (Required)

General structure

```
list of (
string //The preset name to delete
)
```

XML-RPC (PHP structure)

```
[presetnames] =>
Array
(
    [0] => string
)
```

REST (POST parameters)

```
presetnames[0]= string
```

Response

General structure

```

object {
  result int //The processing result
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)
}

```

XML-RPC (PHP structure)

```

Array
(
  [result] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="result">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`mod_data_get_data_access_information`

Return access information for a given database.

Arguments

`databaseid` (Required)

Database instance id.

General structure

```
int //Database instance id.
```

XML-RPC (PHP structure)

```
[databaseid] => int
```

REST (POST parameters)

```
databaseid= int
```

`groupid` (Default to "0")

Group id, 0 means that the function will determine the user group.

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group.
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response

General structure

```

object {
    groupid int //User current group id (calculated)
    canaddentry int //Whether the user can add entries or not.
    canmanageentries int //Whether the user can manage entries or not.
    canapprove int //Whether the user can approve entries or not.
    timeavailable int //Whether the database is available or not by time restrictions.
    inreadonlyperiod int //Whether the database is in read mode only.
    numentries int //The number of entries the current user added.
    entrieslefttoadd int //The number of entries left to complete the activity.
    entrieslefttoview int //The number of entries left to view other users entries.
    warnings Optional //List of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
}
)

```

XML-RPC (PHP structure)

```

Array
(
    [groupid] => int
    [canaddentry] => int
    [canmanageentries] => int
    [canapprove] => int
    [timeavailable] => int
    [inreadonlyperiod] => int
    [numentries] => int
    [entrieslefttoadd] => int
    [entrieslefttoview] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="groupid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canaddentry">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanageentries">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canapprove">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timeavailable">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="inreadonlyperiod">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="numentries">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="entrieslefttoadd">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="entrieslefttoview">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_get_databases_by_courses

Returns a list of database instances in a provided set of courses, if no courses are provided then all the database instances the user has access to will be returned.

Arguments

courseids (Default to 'Array ()')

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
courseids[0]= int
```

Response



General structure

```

object {
databases list of (
object {
id int //Database id
course int //Course id
name string //Database name
intro string //The Database intro
introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
lang string //Forced activity Language
comments int //comments enabled
timeavailablefrom int //timeavailablefrom field
timeavailableto int //timeavailableto field
timeviewfrom int //timeviewfrom field
timeviewto int //timeviewto field
requiredentries int //requiredentries field
requiredentriestoview int //requiredentriestoview field
maxentries int //maxentries field
rssarticles int //rssarticles field
singletemplate string //singletemplate field
listtemplate string //listtemplate field
listtemplateheader string //listtemplateheader field
listtemplatefooter string //listtemplatefooter field
addtemplate string //addtemplate field
rsstemplate string //rsstemplate field
rsstitletemplate string //rsstitletemplate field
csstemplate string //csstemplate field
jstemplate string //jstemplate field
asearchtemplate string //asearchtemplate field
approval int //approval field
manageapproved int //manageapproved field
scale int Optional //scale field
assessed int Optional //assessed field
assesstimestart int Optional //assesstimestart field
assesstimefinish int Optional //assesstimefinish field
defaultsort int //defaultsort field
defaultsortdir int //defaultsortdir field
editany int Optional //editany field (not used any more)
notification int Optional //notification field (not used any more)
timemodified int Optional //Time modified
coursemodule int //coursemodule
introfiles Optional //introfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
})
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
}

```



XML-RPC (PHP structure)

```
  (
    [databases] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [id] => int
                    [course] => int
                    [name] => string
                    [intro] => string
                    [introformat] => int
                    [lang] => string
                    [comments] => int
                    [timeavailablefrom] => int
                    [timeavailableto] => int
                    [timeviewfrom] => int
                    [timeviewto] => int
                    [requireddentries] => int
                    [requiredentriesstoview] => int
                    [maxentries] => int
                    [rssarticles] => int
                    [singletemplate] => string
                    [listtemplate] => string
                    [listtemplateheader] => string
                    [listtemplatefooter] => string
                    [addtemplate] => string
                    [rsstemplate] => string
                    [rsstitemplate] => string
                    [csstemplate] => string
                    [jstemplate] => string
                    [asearchtemplate] => string
                    [approval] => int
                    [manageapproved] => int
                    [scale] => int
                    [assessed] => int
                    [assesstimestart] => int
                    [assesstimefinish] => int
                    [defaultsort] => int
                    [defaultsortdir] => int
                    [editany] => int
                    [notification] => int
                    [timemodified] => int
                    [coursemodule] => int
                    [introfiles] =>
                      Array
                        (
                          [
                            [0] =>
                              Array
                                (
                                  [
                                    [filename] => string
                                    [filepath] => string
                                    [filesize] => int
                                    [fileurl] => string
                                    [timemodified] => int
                                    [mimetype] => string
                                    [isexternalfile] => int
                                    [repositorytype] => string
                                  )
                                )
                          )
                        )
                    )
      )
    )
  )
  [
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
          )
        )
  )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="databases">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="course">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="intro">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="introformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="lang">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="comments">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeavailablefrom">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeavailableto">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeviewfrom">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeviewto">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requiredentries">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="requiredentriesstoview">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="maxentries">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="rssarticles">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="singletemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="listtemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="listtemplateheader">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="listtemplatefooter">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="addtemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="rsstemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="rsstitletemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="csstemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="jstemplate">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="asearchtemplate">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```
<KEY name="approval">
    <VALUE>int</VALUE>
</KEY>
<KEY name="manageapproved">
    <VALUE>int</VALUE>
</KEY>
<KEY name="scale">
    <VALUE>int</VALUE>
</KEY>
<KEY name="assessed">
    <VALUE>int</VALUE>
</KEY>
<KEY name="assessstart">
    <VALUE>int</VALUE>
</KEY>
<KEY name="assessfinish">
    <VALUE>int</VALUE>
</KEY>
<KEY name="defaultsort">
    <VALUE>int</VALUE>
</KEY>
<KEY name="defaultrightdir">
    <VALUE>int</VALUE>
</KEY>
<KEY name="editany">
    <VALUE>int</VALUE>
</KEY>
<KEY name="notification">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="coursemodule">
    <VALUE>int</VALUE>
</KEY>
<KEY name="introfiles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

        </KEY>
    </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_get_entries

Return the complete list of entries of the given database.

Arguments

databaseid (Required)

data instance id

General structure

```
int //data instance id
```

XML-RPC (PHP structure)

```
[databaseid] => int
```

**REST (POST parameters)**

```
databaseid= int
```

groupid (Default to "0")

Group id, 0 means that the function will determine the user group

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

returncontents (Default to "")

Whether to return contents or not. This will return each entry raw contents and the complete list view (using the template).

General structure

```
int Default to "" //Whether to return contents or not. This will return each entry  
raw contents and the complete list view (using the template).
```

XML-RPC (PHP structure)

```
[returncontents] => int
```

REST (POST parameters)

```
returncontents= int
```

sort (Default to "null")

Sort the records by this field id, reserved ids are: 0: timeadded -1: firstname -2: lastname -3: approved -4: timemodified. Empty for using the default database setting.

General structure

```
int Default to "null" //Sort the records by this field id, reserved ids are:  
0: timeadded  
-1: firstname  
-2: lastname  
-3: approved  
-4: timemodified.  
Empty for using the default database setting.
```

XML-RPC (PHP structure)

```
[sort] => int
```

REST (POST parameters)

```
sort= int
```

order (Default to "null")

The direction of the sorting: 'ASC' or 'DESC'. Empty for using the default database setting.

**General structure**

```
string Default to "null" //The direction of the sorting: 'ASC' or 'DESC'.  
Empty for using the default database setting.
```

XML-RPC (PHP structure)

```
[order] => string
```

REST (POST parameters)

```
order= string
```

page (Default to "0")

The page of records to return.

General structure

```
int Default to "0" //The page of records to return.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "0")

The number of records to return per page

General structure

```
int Default to "0" //The number of records to return per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

Response



General structure

```

object {
entries list of (
object {
id int //Record id.
userid int //The id of the user who created the record.
groupid int //The group id this record belongs to (0 for no groups).
dataid int //The database id this record belongs to.
timecreated int //Time the record was created.
timemodified int //Last time the record was modified.
approved int //Whether the entry has been approved (if the database is configured in that way).
canmanageentry int //Whether the current user can manage this entry
fullname string Optional //The user who created the entry fullname.
contents Optional //The record contents.
list of (
object {
id int //Content id.
fieldid int //The field type of the content.
recordid int //The record this content belongs to.
content string //Contents.
content1 string //Contents.
content2 string //Contents.
content3 string //Contents.
content4 string //Contents.
files Optional //files
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)
)tags Optional //Tags.
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)totalcount int //Total count of records.
totalfilesize int //Total size (bytes) of the files included in the records.
listviewcontents string Optional //The List view contents as is rendered in the site.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
}

```




```
[warningcode] => string  
[message] => string  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="entries">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="userid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="groupid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="dataid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="approved">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmanageentry">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="contents">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="fieldid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="recordid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="content">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content1">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content2">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content3">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content4">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="files">
                  <MULTIPLE>
                    <SINGLE>
                      <KEY name="filename">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="filepath">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="filesize">
                        <VALUE>int</VALUE>
                      </KEY>
                      <KEY name="fileurl">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="timemodified">
                        <VALUE>int</VALUE>
                      </KEY>
                      <KEY name="mimetype">
                        <VALUE>string</VALUE>
                      </KEY>

```



```
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="rawname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isstandard">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagcollid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstanceid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstancecontextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="ordering">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="flag">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="totalcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="totalfilesize">
    <VALUE>int</VALUE>
</KEY>
<KEY name="listviewcontents">
    <VALUE>string</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_get_entry

Return one entry record from the database, including contents optionally.

Arguments

entryid (Required)

record entry id

General structure

```
int //record entry id
```

XML-RPC (PHP structure)

```
[entryid] => int
```

REST (POST parameters)

```
entryid= int
```

returncontents (Default to "")

Whether to return contents or not.



General structure

```
int Default to "" //Whether to return contents or not.
```



XML-RPC (PHP structure)

```
[returncontents] => int
```

REST (POST parameters)

```
returncontents= int
```

Response

General structure



```
aggregatestr string Optional //Aggregated ratings as string.  
aggregatelabel string Optional //The aggregation Label.  
count int Optional //Ratings count (used when aggregating).  
rating int Optional //The rating the user gave.  
canrate int Optional //Whether the user can rate the item.  
canviewaggregate int Optional //Whether the user can view the aggregated grade.  
}  
}  
warnings Optional //List of warnings  
list of (  
    warning  
object {  
    item string Optional //item  
    itemid int Optional //item id  
    warningcode string //the warning code can be used by the client app to implement specific behaviour  
    message string //untranslated english message to explain the warning  
}  
})
```



XML-RPC (PHP structure)

```

Array
(
    [entry] =>
        Array
        (
            [id] => int
            [userid] => int
            [groupid] => int
            [dataid] => int
            [timecreated] => int
            [timemodified] => int
            [approved] => int
            [cammanageentry] => int
            [fullname] => string
            [contents] =>
                Array
                (
                    [0] =>
                        Array
                        (
                            [id] => int
                            [fieldid] => int
                            [recordid] => int
                            [content] => string
                            [content1] => string
                            [content2] => string
                            [content3] => string
                            [content4] => string
                            [files] =>
                                Array
                                (
                                    [0] =>
                                        Array
                                        (
                                            [filename] => string
                                            [filepath] => string
                                            [filesize] => int
                                            [fileurl] => string
                                            [timemodified] => int
                                            [mimetype] => string
                                            [isexternalfile] => int
                                            [repositorytype] => string
                                        )
                                )
                        )
                )
            )
        )
    [tags] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [name] => string
                    [rawname] => string
                    [issstandard] => int
                    [tagcollid] => int
                    [taginstanceid] => int
                    [taginstancecontextid] => int
                    [itemid] => int
                    [ordering] => int
                    [flag] => int
                )
        )
    )
)
[entryviewcontents] => string
[ratinginfo] =>
    Array
    (
        [contextid] => int
        [component] => string
        [ratingarea] => string
        [canviewall] => int
        [canviewany] => int
        [scales] =>
            Array
            (
                [0] =>
                    Array
                    (

```



```
[id] => int
[courseid] => int
[name] => string
[max] => int
[isnumeric] => int
[items] =>
    Array
        (
            [0] =>
                Array
                    (
                        [value] => int
                        [name] => string
                    )
        )
)
)
[ratings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [itemid] => int
                        [scaleid] => int
                        [userid] => int
                        [aggregate] => double
                        [aggregatestr] => string
                        [aggregatelabel] => string
                        [count] => int
                        [rating] => int
                        [canrate] => int
                        [canviewaggregate] => int
                    )
            )
        )
)
)
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="entry">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="dataid">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timecreated">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="approved">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="canmanageentry">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="fullname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="contents">
          <MULTIPLE>
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="fieldid">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="recordid">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="content">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="content1">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="content2">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="content3">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="content4">
                <VALUE>string</VALUE>
              </KEY>
            <KEY name="files">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="filename">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="filepath">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="filesize">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="fileurl">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="timemodified">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="mimetype">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="isexternalfile">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="rawname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isstandard">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagcollid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstanceid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstancecontextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="ordering">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="flag">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="entryviewcontents">
    <VALUE>string</VALUE>
</KEY>
<KEY name="ratinginfo">
    <SINGLE>
        <KEY name="contextid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="component">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="ratingarea">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="canviewall">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewany">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="scales">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="max">
                        <VALUE>int</VALUE>

```



```
</KEY>
<KEY name="isnumeric">
    <VALUE>int</VALUE>
</KEY>
<KEY name="items">
    <MULTIPLE>
        <SINGLE>
            <KEY name="value">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="ratings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="scaleid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="aggregate">
                <VALUE>double</VALUE>
            </KEY>
            <KEY name="aggregatestr">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aggregatelabel">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="count">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="rating">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="canrate">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="canviewaggregate">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_get_fields

Return the list of configured fields for the given database.

Arguments**databaseid** (Required)

Database instance id.

General structure

```
int //Database instance id.
```

XML-RPC (PHP structure)

```
[databaseid] => int
```

REST (POST parameters)

```
databaseid= int
```

Response

General structure

```

object {
fields list of (
object {
id int    //Field id.
dataid int   //The field type of the content.
type string   //The field type.
name string   //The field name.
description string   //The field description.
required int   //Whether is a field required or not.
param1 string   //Field parameters
param2 string   //Field parameters
param3 string   //Field parameters
param4 string   //Field parameters
param5 string   //Field parameters
param6 string   //Field parameters
param7 string   //Field parameters
param8 string   //Field parameters
param9 string   //Field parameters
param10 string  //Field parameters
}
)warnings  Optional //List of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)}
)

```

XML-RPC (PHP structure)

```

Array
(
[fields] =>
  Array
  (
    [0] =>
      Array
      (
        [
          [id] => int
          [dataid] => int
          [type] => string
          [name] => string
          [description] => string
          [required] => int
          [param1] => string
          [param2] => string
          [param3] => string
          [param4] => string
          [param5] => string
          [param6] => string
          [param7] => string
          [param8] => string
          [param9] => string
          [param10] => string
        )
      )
    [warnings] =>
      Array
      (
        [0] =>
          Array
          (
            [
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
          )
      )
    )
  )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="fields">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dataid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="type">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="description">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="required">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="param1">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param2">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param3">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param4">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param5">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param6">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param7">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param8">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param9">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="param10">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_get_mapping_information

Get importing information

Arguments

cmid (Required)

Id of the data activity

General structure

```
int //Id of the data activity
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

importedpreset (Required)

Preset to be imported



General structure

```
string //Preset to be imported
```

XML-RPC (PHP structure)

```
[importedpreset] => string
```

REST (POST parameters)

```
importedpreset= string
```

Response

General structure

```
object {
  data Optional //Information to import if everything went fine
  object {
    needsmapping int //Whether the importing needs mapping or not
    presetname string //Name of the applied preset
    fieldstocreate string //List of field names to create
    fieldstoremove string //List of field names to remove
  }
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [data] =>
  Array
  (
    [needsmapping] => int
    [presetname] => string
    [fieldstocreate] => string
    [fieldstoremove] => string
  )
  [warnings] =>
  Array
  (
    [0] =>
    Array
    (
      [item] => string
      [itemid] => int
      [warningcode] => string
      [message] => string
    )
  )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="data">
      <SINGLE>
        <KEY name="needsmapping">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="presetname">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fieldstocreate">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="fieldstoremove">
          <VALUE>string</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_data_search_entries

Search for entries in the given database.

Arguments**databaseid** (Required)

data instance id

General structure

```
int //data instance id
```

XML-RPC (PHP structure)

```
[databaseid] => int
```

REST (POST parameters)

```
databaseid= int
```

groupid (Default to "0")

Group id, 0 means that the function will determine the user group

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

returncontents (Default to "")

Whether to return contents or not.

General structure

```
int Default to "" //Whether to return contents or not.
```

**search** (Default to "")

search string (empty when using advanced)

General structure

```
string Default to "" //search string (empty when using advanced)
```

XML-RPC (PHP structure)

```
[search] => string
```

REST (POST parameters)

```
search= string
```

advsearch (Default to "Array ()")

Advanced search

```
General structure

Default to "Array
(
)
" //Advanced search
list of (
object {
name string //Field key for search.
value string //JSON encoded value for search
}
)
```

Use fn or ln for first or last name

XML-RPC (PHP structure)

```
[advsearch] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

REST (POST parameters)

```
advsearch[0][name]= string
advsearch[0][value]= string
```

sort (Default to "null")

Sort the records by this field id, reserved ids are: 0: timeadded -1: firstname -2: lastname -3: approved -4: timemodified. Empty for using the default database setting.

General structure

```
int Default to "null" //Sort the records by this field id, reserved ids are:
0: timeadded
-1: firstname
-2: lastname
-3: approved
-4: timemodified.
Empty for using the default database setting.
```

XML-RPC (PHP structure)

```
[sort] => int
```

**REST (POST parameters)**

```
sort= int
```

order (Default to "null")

The direction of the sorting: 'ASC' or 'DESC'. Empty for using the default database setting.

General structure

```
string Default to "null" //The direction of the sorting: 'ASC' or 'DESC'.
Empty for using the default database setting.
```

XML-RPC (PHP structure)

```
[order] => string
```

REST (POST parameters)`order= string``page` (Default to "0")

The page of records to return.

General structure

```
int Default to "0" //The page of records to return.
```

XML-RPC (PHP structure)`[page] => int`**REST (POST parameters)**`page= int``perpage` (Default to "0")

The number of records to return per page

General structure

```
int Default to "0" //The number of records to return per page
```

XML-RPC (PHP structure)`[perpage] => int`**REST (POST parameters)**`perpage= int`**Response**

General structure

```

object {
entries list of (
object {
id int //Record id.
userid int //The id of the user who created the record.
groupid int //The group id this record belongs to (0 for no groups).
dataid int //The database id this record belongs to.
timecreated int //Time the record was created.
timemodified int //Last time the record was modified.
approved int //Whether the entry has been approved (if the database is configured in that way).
canmanageentry int //Whether the current user can manage this entry
fullname string Optional //The user who created the entry fullname.
contents Optional //The record contents.
list of (
object {
id int //Content id.
fieldid int //The field type of the content.
recordid int //The record this content belongs to.
content string //Contents.
content1 string //Contents.
content2 string //Contents.
content3 string //Contents.
content4 string //Contents.
files Optional //files
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)
)
tags Optional //Tags.
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)
totalcount int //Total count of records returned by the search.
maxcount int Optional //Total count of records that the user could see in the database
// (if all the search criterias were removed).
listviewcontents string Optional //The List view contents as is rendered in the site.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
}

```




```
[warningcode] => string  
[message] => string  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="entries">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="userid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="groupid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="dataid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timecreated">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="approved">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="canmanageentry">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="contents">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="fieldid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="recordid">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="content">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content1">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content2">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content3">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="content4">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="files">
                  <MULTIPLE>
                    <SINGLE>
                      <KEY name="filename">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="filepath">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="filesize">
                        <VALUE>int</VALUE>
                      </KEY>
                      <KEY name="fileurl">
                        <VALUE>string</VALUE>
                      </KEY>
                      <KEY name="timemodified">
                        <VALUE>int</VALUE>
                      </KEY>
                      <KEY name="mimetype">
                        <VALUE>string</VALUE>
                      </KEY>

```



```
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="rawname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isstandard">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagcollid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstanceid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="taginstancecontextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="ordering">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="flag">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="totalcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="maxcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="listviewcontents">
    <VALUE>string</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_update_entry

Updates an existing entry.

Arguments

entryid (Required)

The entry record id.

General structure

```
int //The entry record id.
```

XML-RPC (PHP structure)

```
[entryid] => int
```

REST (POST parameters)

```
entryid= int
```

data (Required)

The fields data to be updated

General structure

```
//The fields data to be updated
list of (
object {
fieldid int //The field id.
subfield string Default to "null" //The subfield name (if required).
value string //The new contents for the field always JSON encoded.
})
```

XML-RPC (PHP structure)

```
[data] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [fieldid] => int
                        [subfield] => string
                        [value] => string
                    )
                )
        )
    )
```

REST (POST parameters)

```
data[0][fieldid]= int
data[0][subfield]= string
data[0][value]= string
```

Response**General structure**

```
object {
    updated int //True if the entry was successfully updated, false other wise.
    generalnotifications list of (
        string //General notifications
    )fieldnotifications list of (
        object {
            fieldname string //The field name.
            notification string //The notification for the field.
        }
    )warnings Optional //List of warnings
    list of (
        //warning
    )object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```



XML-RPC (PHP structure)

```
Array
(
    [updated] => int
    [generalnotifications] =>
        Array
            (
                [0] => string
            )
    [fieldnotifications] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [fieldname] => string
                            [notification] => string
                        )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="updated">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="generalnotifications">
            <MULTIPLE>
                <VALUE>string</VALUE>
            </MULTIPLE>
        </KEY>
        <KEY name="fieldnotifications">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="fieldname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="notification">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_data_view_database

Simulate the view.php web interface data: trigger events, completion, etc...

Arguments

databaseid (Required)
data instance id

General structure

```
int //data instance id
```

XML-RPC (PHP structure)

```
[databaseid] => int
```

REST (POST parameters)

```
databaseid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_feedback_get_analysis**

Retrieves the feedback analysis.

Arguments**feedbackid** (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)`feedbackid= int`**groupid** (Default to "0")

Group id, 0 means that the function will determine the user group

General structure`int Default to "0" //Group id, 0 means that the function will determine the user group`**XML-RPC (PHP structure)**`[groupid] => int`**REST (POST parameters)**`groupid= int`**courseid** (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure`int Default to "0" //Course where user completes the feedback (for site feedbacks only).`**XML-RPC (PHP structure)**`[courseid] => int`**REST (POST parameters)**`courseid= int`**Response**

```
General structure

object {
    completedcount int //Number of completed submissions.
    itemscount int //Number of items (questions).
    itemsdata list of (
        object {
            item object {
                id int //The record id.
                feedback int //The feedback instance id this records belongs to.
                template int //If it belongs to a template, the template id.
                name string //The item name.
                label string //The item label.
                presentation string //The text describing the item or the available possible answers.
                typ string //The type of the item.
                hasvalue int //Whether it has a value or not.
                position int //The position in the List of questions.
                required int //Whether is a item (question) required or not.
                dependitem int //The item id this item depend on.
                dependvalue string //The depend value.
                options string //Different additional settings for the item (question).
                itemfiles //itemfiles
            }
            list of (
                object {
                    contextid int //contextid
                    component string //component
                    filearea string //filearea
                    itemid int //itemid
                    filepath string //filepath
                    filename string //filename
                    isdir int //isdir
                    isimage int //isimage
                    timemodified int //timemodified
                    timecreated int //timecreated
                    filesize int //filesize
                    author string //author
                    license string //license
                    filenameshort string //filenameshort
                    filesizeformatted string //filesizeformatted
                    icon string //icon
                    timecreatedformatted string //timecreatedformatted
                    timemodifiedformatted string //timemodifiedformatted
                    url string //url
                }
            )
        )itemnumber int //The item position number
        otherdata string //Additional data that may be required by external functions
    }
    data list of (
        string //The analysis data (can be json encoded)
    )
    warnings Optional //List of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )
}
```



XML-RPC (PHP structure)

```

Array
(
    [
        [completedcount] => int
        [itemscount] => int
        [itemsdata] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] =>
                                    Array
                                    (
                                        [
                                            [id] => int
                                            [feedback] => int
                                            [template] => int
                                            [name] => string
                                            [label] => string
                                            [presentation] => string
                                            [typ] => string
                                            [hasvalue] => int
                                            [position] => int
                                            [required] => int
                                            [dependitem] => int
                                            [dependvalue] => string
                                            [options] => string
                                            [itemfiles] =>
                                                Array
                                                (
                                                    [
                                                        [0] =>
                                                            Array
                                                            (
                                                                [
                                                                    [
                                                                        [contextid] => int
                                                                        [component] => string
                                                                        [filearea] => string
                                                                        [itemid] => int
                                                                        [filepath] => string
                                                                        [filename] => string
                                                                        [isdir] => int
                                                                        [isimage] => int
                                                                        [timemodified] => int
                                                                        [timecreated] => int
                                                                        [filesize] => int
                                                                        [author] => string
                                                                        [license] => string
                                                                        [filenameshort] => string
                                                                        [filesizeformatted] => string
                                                                        [icon] => string
                                                                        [timecreatedformatted] => string
                                                                        [timemodifiedformatted] => string
                                                                        [url] => string
                                                                    )
                                                                )
                                                        )
                                                    )
                                                )
                                            [itemnumber] => int
                                            [otherdata] => string
                                        )
                                    )
                                )
                            )
                        )
                    [data] =>
                        Array
                        (
                            [
                                [0] => string
                            )
                        )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="completedcount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="itemscount">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="itemsdata">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="feedback">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="template">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="name">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="label">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="presentation">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="typ">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="hasvalue">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="position">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="required">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="dependitem">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="dependvalue">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="options">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="itemfiles">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="contextid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="component">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filearea">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="filepath">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filename">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isdir">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="isimage">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filesize">
    <VALUE>int</VALUE>
</KEY>
<KEY name="author">
    <VALUE>string</VALUE>
</KEY>
<KEY name="license">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="itemnumber">
    <VALUE>int</VALUE>
</KEY>
<KEY name="otherdata">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="data">
    <MULTIPLE>
        <VALUE>string</VALUE>
    </MULTIPLE>
    </KEY>
    </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</RESPONSE>

```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_current_completed_tmp

Returns the temporary completion record for the current user.

Arguments**feedbackid** (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

**XML-RPC (PHP structure)**

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```

object {
feedback object {
id int //The record id.
feedback int //The feedback instance id this records belongs to.
userid int //The user who completed the feedback (0 for anonymous).
guestid string //For guests, this is the session key.
timemodified int //The last time the feedback was completed.
random_response int //The response number (used when shuffling anonymous responses).
anonymous_response int //Whether is an anonymous response.
courseid int //The course id where the feedback was completed.
}
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
}

```

XML-RPC (PHP structure)

```

Array
(
[feedback] =>
    Array
    (
        [
            [id] => int
            [feedback] => int
            [userid] => int
            [guestid] => string
            [timemodified] => int
            [random_response] => int
            [anonymous_response] => int
            [courseid] => int
        )
[warnings] =>
    Array
    (
        [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
        )
    )
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="feedback">
            <SINGLE>
                <KEY name="id">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="feedback">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="userid">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="guestid">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="timemodified">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="random_response">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="anonymous_response">
                    <VALUE>int</VALUE>
                </KEY>
                <KEY name="courseid">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_feedback_access_information

Return access information for a given feedback.

Arguments**feedbackid** (Required)

Feedback instance id.

General structure

```
int //Feedback instance id.
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
    canviewanalysis int //Whether the user can view the analysis or not.
    cancomplete int //Whether the user can complete the feedback or not.
    cansubmit int //Whether the user can submit the feedback or not.
    candeletesubmissions int //Whether the user can delete submissions or not.
    canviewreports int //Whether the user can view the feedback reports or not.
    canedititems int //Whether the user can edit feedback items or not.
    isempty int //Whether the feedback has questions or not.
    isopen int //Whether the feedback has active access time restrictions or not.
    isalreadysubmitted int //Whether the feedback is already submitted or not.
    isanonymous int //Whether the feedback is anonymous or not.
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [canviewanalysis] => int
    [cancomplete] => int
    [cansubmit] => int
    [candeletesubmissions] => int
    [canviewreports] => int
    [canedititems] => int
    [isempty] => int
    [isopen] => int
    [isalreadysubmitted] => int
    [isanonymous] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="canviewanalysis">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="cancomplete">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="cansubmit">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="candeletesubmissions">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canviewreports">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="canedititems">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isempty">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isopen">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isalreadysubmitted">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isanonymous">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_feedbacks_by_courses

Returns a list of feedbacks in a provided list of courses, if no list is provided all feedbacks that the user can view will be returned.

Arguments

courseids (Default to 'Array ()')

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //Course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
courseids[0]= int
```

Response



General structure

```
object {
  feedbacks list of (
    object {
      id int //The primary key of the record.
      course int //Course id this feedback is part of.
      name string //Feedback name.
      intro string //Feedback introduction text.
      introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      lang string //Forced activity Language
      anonymous int //Whether the feedback is anonymous.
      email_notification int Optional //Whether email notifications will be sent to teachers.
      multiple_submit int //Whether multiple submissions are allowed.
      autonumbering int //Whether questions should be auto-numbered.
      site_after_submit string Optional //Link to next page after submission.
      page_after_submit string Optional //Text to display after submission.
      page_after_submitformat int Default to "1" //page_after_submit format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
      publish_stats int //Whether stats should be published.
      timeopen int Optional //Allow answers from this time.
      timelimit int Optional //Allow answers until this time.
      timemodified int Optional //The time this record was modified.
      completionsubmit int //If this field is set to 1, then the activity will be automatically marked as complete on submission.
      coursemodule int //coursemodule
      introfiles //introfiles
      list of (
        object {
          filename string Optional //File name.
          filepath string Optional //File path.
          filesize int Optional //File size.
          fileurl string Optional //Downloadable file url.
          timemodified int Optional //Time modified.
          mimetype string Optional //File mime type.
          isexternalfile int Optional //Whether is an external file.
          repositorytype string Optional //The repository type for the external files.
        }
      )pageaftersubmitfiles Optional //pageaftersubmitfiles
      list of (
        object {
          filename string Optional //File name.
          filepath string Optional //File path.
          filesize int Optional //File size.
          fileurl string Optional //Downloadable file url.
          timemodified int Optional //Time modified.
          mimetype string Optional //File mime type.
          isexternalfile int Optional //Whether is an external file.
          repositorytype string Optional //The repository type for the external files.
        }
      )
    })
  )warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```



XML-RPC (PHP structure)

```

Array
(
    [
        [feedbacks] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [course] => int
                                [name] => string
                                [intro] => string
                                [introformat] => int
                                [lang] => string
                                [anonymous] => int
                                [email_notification] => int
                                [multiple_submit] => int
                                [autonumbering] => int
                                [site_after_submit] => string
                                [page_after_submit] => string
                                [page_after_submitformat] => int
                                [publish_stats] => int
                                [timeopen] => int
                                [timeclose] => int
                                [timemodified] => int
                                [completionsubmit] => int
                                [coursemodule] => int
                                [introfiles] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [filename] => string
                                                        [filepath] => string
                                                        [filesize] => int
                                                        [fileurl] => string
                                                        [timemodified] => int
                                                        [mimetype] => string
                                                        [isexternalfile] => int
                                                        [repositorytype] => string
                                                    )
                                                )
                                        )
                                    )
                                [pageaftersubmitfiles] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [filename] => string
                                                        [filepath] => string
                                                        [filesize] => int
                                                        [fileurl] => string
                                                        [timemodified] => int
                                                        [mimetype] => string
                                                        [isexternalfile] => int
                                                        [repositorytype] => string
                                                    )
                                                )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        [warnings] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                        )
                )
            )
        )
    )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="feedbacks">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="course">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="name">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="intro">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="introformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="lang">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="anonymous">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="email_notification">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="multiple_submit">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="autonumbering">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="site_after_submit">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="page_after_submit">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="page_after_submitformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="publish_stats">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeopen">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timeclose">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="completionsubmit">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="coursemodule">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="introfiles">
            <MULTIPLE>
              <SINGLE>
                <KEY name="filename">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filepath">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="filesize">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="fileurl">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="timemodified">
                  <VALUE>int</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```

<KEY name="mimetype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
    <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="pageaftersubmitfiles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
            </SINGLE>
        </MULTIPLE>
    </KEY>
    <SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_finished_responses

Retrieves responses from the last finished attempt.

Arguments

feedbackid (Required)

Feedback instance id.

General structure

```
int //Feedback instance id.
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```

object {
  responses list of (
    object {
      id int //The record id.
      course_id int //The course id this record belongs to.
      item int //The item id that was responded.
      completed int //Reference to the feedback_completed table.
      tmp_completed int //Old field - not used anymore.
      value string //The response value.
    }
  ) warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}

```

XML-RPC (PHP structure)

```

Array
(
  [responses] =>
    Array
      (
        [0] =>
          Array
            (
              [id] => int
              [course_id] => int
              [item] => int
              [completed] => int
              [tmp_completed] => int
              [value] => string
            )
      )
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="responses">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="course_id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="item">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="completed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="tmp_completed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_items

Returns the items (questions) in the given feedback.

Arguments

feedbackid (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response



General structure

```
object {
  items list of (
    object {
      id int //The record id.
      feedback int //The feedback instance id this records belongs to.
      template int //If it belongs to a template, the template id.
      name string //The item name.
      label string //The item label.
      presentation string //The text describing the item or the available possible answers.
      typ string //The type of the item.
      hasvalue int //Whether it has a value or not.
      position int //The position in the List of questions.
      required int //Whether is a item (question) required or not.
      dependitem int //The item id this item depend on.
      dependvalue string //The depend value.
      options string //Different additional settings for the item (question).
      itemfiles //itemfiles
      list of (
        object {
          contextid int //contextid
          component string //component
          filearea string //filearea
          itemid int //itemid
          filepath string //filepath
          filename string //filename
          isdir int //isdir
          isimage int //isimage
          timemodified int //timemodified
          timecreated int //timecreated
          filesize int //filesize
          author string //author
          license string //license
          filenameshort string //filenameshort
          filesizeformatted string //filesizeformatted
          icon string //icon
          timecreatedformatted string //timecreatedformatted
          timemodifiedformatted string //timemodifiedformatted
          url string //url
        }
      )
      itemnumber int //The item position number
      otherdata string //Additional data that may be required by external functions
    }
  )
  warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
    [items] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [feedback] => int
                            [template] => int
                            [name] => string
                            [label] => string
                            [presentation] => string
                            [typ] => string
                            [hasvalue] => int
                            [position] => int
                            [required] => int
                            [dependitem] => int
                            [dependvalue] => string
                            [options] => string
                            [itemfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [
                                                        [contextid] => int
                                                        [component] => string
                                                        [filearea] => string
                                                        [itemid] => int
                                                        [filepath] => string
                                                        [filename] => string
                                                        [isdir] => int
                                                        [isimage] => int
                                                        [timemodified] => int
                                                        [timecreated] => int
                                                        [filesize] => int
                                                        [author] => string
                                                        [license] => string
                                                        [filenameshort] => string
                                                        [filesizeformatted] => string
                                                        [icon] => string
                                                        [timecreatedformatted] => string
                                                        [timemodifiedformatted] => string
                                                        [url] => string
                                                    )
                                                )
                                        )
                                    )
                            [itemnumber] => int
                            [otherdata] => string
                        )
                )
            )
        [warnings] =>
            Array
                (
                    [0] =>
                        Array
                            (
                                [item] => string
                                [itemid] => int
                                [warningcode] => string
                                [message] => string
                            )
                )
            )
        )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="items">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="feedback">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="template">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="label">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="presentation">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="typ">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="hasvalue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="position">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="required">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dependitem">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dependvalue">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="options">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="contextid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="component">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filearea">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="itemid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isdir">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="isimage">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```

<KEY name="author">
    <VALUE>string</VALUE>
</KEY>
<KEY name="license">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="itemnumber">
    <VALUE>int</VALUE>
</KEY>
<KEY name="otherdata">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_last_completed

Retrieves the last completion record for the current user.

Arguments

feedbackid (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
object {
    completed object {
        id int //The record id.
        feedback int //The feedback instance id this records belongs to.
        userid int //The user who completed the feedback (0 for anonymous).
        timemodified int //The last time the feedback was completed.
        random_response int //The response number (used when shuffling anonymous responses).
        anonymous_response int //Whether is an anonymous response.
        courseid int //The course id where the feedback was completed.
    }
    warnings Optional //List of warnings
    list of (
        //warning
    object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)}
```

XML-RPC (PHP structure)

```
Array
(
    [completed] =>
        Array
            (
                [id] => int
                [feedback] => int
                [userid] => int
                [timemodified] => int
                [random_response] => int
                [anonymous_response] => int
                [courseid] => int
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <KEY name="completed">
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="feedback">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="userid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="random_response">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="anonymous_response">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="courseid">
            <VALUE>int</VALUE>
        </KEY>
    </KEY>
    <KEY name="warnings">
        <MULTIPLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </KEY>
    </MULTIPLE>
    </KEY>
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_non_respondents

Retrieves a list of students who didn't submit the feedback.

Arguments

feedbackid (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

groupid (Default to "0")

Group id, 0 means that the function will determine the user group.

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group.
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

sort (Default to "lastaccess")

Sort param, must be firstname, lastname or lastaccess (default).

General structure

```
string Default to "lastaccess" //Sort param, must be firstname, Lastname or Lastaccess (default).
```



XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

page (Default to "0")

The page of records to return.

General structure

```
int Default to "0" //The page of records to return.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "0")

The number of records to return per page.

General structure

```
int Default to "0" //The number of records to return per page.
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
  users list of (
    object {
      courseid int //Course id
      userid int //The user id
      fullname string //User full name
      started int //If the user has started the attempt
    }
    )total int //Total number of non respondents
    warnings Optional //List of warnings
    list of (
      //warning
      object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
      }
    )
}
```

XML-RPC (PHP structure)

```

Array
(
    [
        [users] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [courseid] => int
                                [userid] => int
                                [fullname] => string
                                [started] => int
                            )
                        )
                ]
                [
                    [total] => int
                ]
                [
                    [warnings] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [item] => string
                                            [itemid] => int
                                            [warningcode] => string
                                            [message] => string
                                        )
                                    )
                            ]
                        )
                ]
            )
    )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="users">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="courseid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="started">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="total">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```



Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_page_items

Get a single feedback page items.

Arguments**feedbackid (Required)**

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

page (Required)

The page to get starting by 0

**General structure**

```
int //The page to get starting by 0
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
  items list of (
    object {
      id int //The record id.
      feedback int //The feedback instance id this records belongs to.
      template int //If it belongs to a template, the template id.
      name string //The item name.
      label string //The item label.
      presentation string //The text describing the item or the available possible answers.
      typ string //The type of the item.
      hasvalue int //Whether it has a value or not.
      position int //The position in the list of questions.
      required int //Whether is a item (question) required or not.
      dependitem int //The item id this item depend on.
      dependvalue string //The depend value.
      options string //Different additional settings for the item (question).
      itemfiles //itemfiles
      list of (
        object {
          contextid int //contextid
          component string //component
          filearea string //filearea
          itemid int //itemid
          filepath string //filepath
          filename string //filename
          isdir int //isdir
          isimage int //isimage
          timemodified int //timemodified
          timecreated int //timecreated
          filesize int //filesize
          author string //author
          license string //license
          filenameshort string //filenameshort
          filesizeformatted string //filesizeformatted
          icon string //icon
          timecreatedformatted string //timecreatedformatted
          timemodifiedformatted string //timemodifiedformatted
          url string //url
        }
      )itemnumber int //The item position number
      otherdata string //Additional data that may be required by external functions
    }
  )hasprevpage int //Whether is a previous page.
  hasnextpage int //Whether there are more pages.
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
  )}
```



XML-RPC (PHP structure)

```

Array
(
    [items] =>
        Array
        (
            [0] =>
                Array
                (
                    [id] => int
                    [feedback] => int
                    [template] => int
                    [name] => string
                    [label] => string
                    [presentation] => string
                    [typ] => string
                    [hasvalue] => int
                    [position] => int
                    [required] => int
                    [dependitem] => int
                    [dependvalue] => string
                    [options] => string
                    [itemfiles] =>
                        Array
                        (
                            [0] =>
                                Array
                                (
                                    [
                                        [contextid] => int
                                        [component] => string
                                        [filearea] => string
                                        [itemid] => int
                                        [filepath] => string
                                        [filename] => string
                                        [isdir] => int
                                        [isimage] => int
                                        [timemodified] => int
                                        [timecreated] => int
                                        [filesize] => int
                                        [author] => string
                                        [license] => string
                                        [filenameshort] => string
                                        [filesizeformatted] => string
                                        [icon] => string
                                        [timecreatedformatted] => string
                                        [timemodifiedformatted] => string
                                        [url] => string
                                    )
                                )
                            )
                        )
                    [itemnumber] => int
                    [otherdata] => string
                )
            )
        )
    [hasprevpage] => int
    [hasnextpage] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
            )
        )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="items">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="feedback">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="template">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="label">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="presentation">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="typ">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="hasvalue">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="position">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="required">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dependitem">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="dependvalue">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="options">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="contextid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="component">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filearea">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="itemid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isdir">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="isimage">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="timecreated">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```

<KEY name="author">
    <VALUE>string</VALUE>
</KEY>
<KEY name="license">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="itemnumber">
    <VALUE>int</VALUE>
</KEY>
<KEY name="otherdata">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="hasprevpage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasnextpage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
        </MULTIPLE>
    </KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</RESPONSE>

```



Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_get_responses_analysis

Return the feedback user responses analysis.

Arguments**feedbackid** (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

groupid (Default to "0")

Group id, 0 means that the function will determine the user group

General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

**page** (Default to "0")

The page of records to return.

General structure

```
int Default to "0" //The page of records to return.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

perpage (Default to "0")

The number of records to return per page

General structure

```
int Default to "0" //The number of records to return per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
object {
attempts list of (
object {
id int //Completed id
courseid int //Course id
userid int //User who responded
timemodified int //Time modified for the response
fullname string //User full name
responses list of (
object {
id int //Response id
name string //Response name
printval string //Response ready for output
rawval string //Response raw value
}
)
})
totalattempts int //Total responses count.
anonattempts list of (
object {
id int //Completed id
courseid int //Course id
number int //Response number
responses list of (
object {
id int //Response id
name string //Response name
printval string //Response ready for output
rawval string //Response raw value
}
)
))
totalanonattempts int //Total anonymous responses count.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})
```



XML-RPC (PHP structure)

```

Array
(
    [
        [attempts] =>
            Array
            (
                [
                    [0] =>
                        Array
                        (
                            [
                                [id] => int
                                [courseid] => int
                                [userid] => int
                                [timemodified] => int
                                [fullname] => string
                                [responses] =>
                                    Array
                                    (
                                        [
                                            [0] =>
                                                Array
                                                (
                                                    [
                                                        [id] => int
                                                        [name] => string
                                                        [printval] => string
                                                        [rawval] => string
                                                    )
                                                )
                                        )
                                    )
                )
            )
        [
            [totalattempts] => int
            [anonattempts] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [id] => int
                                    [courseid] => int
                                    [number] => int
                                    [responses] =>
                                        Array
                                        (
                                            [
                                                [0] =>
                                                    Array
                                                    (
                                                        [
                                                            [id] => int
                                                            [name] => string
                                                            [printval] => string
                                                            [rawval] => string
                                                        )
                                                    )
                                            )
                                        )
                                )
                )
            )
        [
            [totalanonattempts] => int
            [warnings] =>
                Array
                (
                    [
                        [0] =>
                            Array
                            (
                                [
                                    [item] => string
                                    [itemid] => int
                                    [warningcode] => string
                                    [message] => string
                                )
                            )
                    )
                )
            )
        )
    )
)

```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="attempts">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="courseid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="userid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="timemodified">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="fullname">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="responses">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="printval">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="rawval">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="totalattempts">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="anonattempts">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="courseid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="number">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="responses">
            <MULTIPLE>
              <SINGLE>
                <KEY name="id">
                  <VALUE>int</VALUE>
                </KEY>
                <KEY name="name">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="printval">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="rawval">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </MULTIPLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="totalanonattempts">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</MULTIPLE>
</KEY>
```



```

<KEY name="warnings">
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_feedback_get_unfinished_responses`

Retrieves responses from the current unfinished attempt.



Arguments

`feedbackid` (Required)

Feedback instance id.

General structure

```
int //Feedback instance id.
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

`courseid` (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
  responses list of (
    object {
      id int //The record id.
      course_id int //The course id this record belongs to.
      item int //The item id that was responded.
      completed int //Reference to the feedback_completedtmp table.
      tmp_completed int //Old field - not used anymore.
      value string //The response value.
    }
  )warnings Optional //List of warnings
  list of (
    //warning
    object {
      item string Optional //item
      itemid int Optional //item id
      warningcode string //the warning code can be used by the client app to implement specific behaviour
      message string //untranslated english message to explain the warning
    }
  )
}
```

XML-RPC (PHP structure)

```
Array
(
  [responses] =>
    Array
      (
        [0] =>
          Array
            (
              [id] => int
              [course_id] => int
              [item] => int
              [completed] => int
              [tmp_completed] => int
              [value] => string
            )
      )
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="responses">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="course_id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="item">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="completed">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="tmp_completed">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="value">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_launch_feedback

Starts or continues a feedback submission.

Arguments

feedbackid (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```
object {
  gopage int //The next page to go (-1 if we were already in the last page). 0 for first page.
  warnings Optional //List of warnings
  list of (
    //warning
  ) object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
}
```

XML-RPC (PHP structure)

```
Array
  (
    [gopage] => int
    [warnings] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                  )
                )
        )
  )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="gopage">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_process_page

Process a jump between pages.

Arguments

feedbackid (Required)

Feedback instance id.

General structure

```
int //Feedback instance id.
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

page (Required)

The page being processed.

General structure

```
int //The page being processed.
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```

responses (Default to "Array ()")

The data to be processed.

General structure

```
Default to "Array
(
)
" //The data to be processed.
list of (
object {
name string //The response name (usually type[index].id).
value string //The response value.
}
)
```

**XML-RPC (PHP structure)**

```
[responses] =>
  Array
    (
      [0] =>
        Array
          (
            [name] => string
            [value] => string
          )
    )
)
```

REST (POST parameters)

```
responses[0][name]= string
responses[0][value]= string
```

goprevious (Default to "")

Whether we want to jump to previous page.

General structure

```
int Default to "" //Whether we want to jump to previous page.
```

XML-RPC (PHP structure)

```
[goprevious] => int
```

REST (POST parameters)

```
goprevious= int
```

courseid (Default to "o")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response**General structure**

```
object {
  jumpto int //The page to jump to.
  completed int //If the user completed the feedback.
  completionpagecontents string //The completion page contents.
  siteaftersubmit string //The link (could be relative) to show after submit.
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
  }
  warningcode string //the warning code can be used by the client app to implement specific behaviour
  message string //untranslated english message to explain the warning
}
}
```

XML-RPC (PHP structure)

```

Array
(
    [jumpto] => int
    [completed] => int
    [completionpagecontents] => string
    [siteaftersubmit] => string
    [warnings] =>
        Array
        (
            [
                [0] =>
                    Array
                    (
                        [
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
                    )
            )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="jumpto">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="completed">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="completionpagecontents">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="siteaftersubmit">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_feedback_view_feedback

Trigger the course module viewed event and update the module completion status.

Arguments

feedbackid (Required)

Feedback instance id

General structure

```
int //Feedback instance id
```

XML-RPC (PHP structure)

```
[feedbackid] => int
```

REST (POST parameters)

```
feedbackid= int
```

moduleviewed (Default to "")

If we need to mark the module as viewed for completion

General structure

```
int Default to "" //If we need to mark the module as viewed for completion
```

XML-RPC (PHP structure)

```
[moduleviewed] => int
```

REST (POST parameters)

```
moduleviewed= int
```



courseid (Default to "0")

Course where user completes the feedback (for site feedbacks only).

General structure

```
int Default to "0" //Course where user completes the feedback (for site feedbacks only).
```

XML-RPC (PHP structure)

```
[courseid] => int
```

REST (POST parameters)

```
courseid= int
```

Response

General structure

```

object {
    status int    //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    object {
        item string  Optional //item
        itemid int   Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_folder_get_folders_by_courses`

Returns a list of folders in a provided list of courses, if no list is provided all folders that the user can view will be returned. Please note that this WS is not returning the folder contents.

Arguments

`courseids` (Default to "Array ()")

Array of course ids

General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int  //Course id
)
```

XML-RPC (PHP structure)

```
[courseids] =>
    Array
        (
            [0] => int
        )
```

REST (POST parameters)

```
courseids[0]= int
```

Response



General structure

```
object {
folders list of (
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
revision int //Incremented when after each file changes, to avoid cache
timemodified int //Last time the folder was modified
display int //Display type of folder contents on a separate page or inline
showexpanded int //1 = expanded, 0 = collapsed for sub-folders
showdownloadfolder int //Whether to show the download folder button
forcedownload int //Whether file download is forced
})
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```



XML-RPC (PHP structure)

```

Array
(
    [folders] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [coursemodule] => int
                            [course] => int
                            [name] => string
                            [intro] => string
                            [introformat] => int
                            [introfiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            [section] => int
                            [visible] => int
                            [groupmode] => int
                            [groupingid] => int
                            [lang] => string
                            [revision] => int
                            [timemodified] => int
                            [display] => int
                            [showexpanded] => int
                            [showdownloadfolder] => int
                            [forcedownload] => int
                        )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
        )
)

```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="folders">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="coursemodule">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="course">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="intro">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="introformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="introfiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="repositorytype">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="section">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupmode">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="groupingid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="lang">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="revision">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="timemodified">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="display">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="showexpanded">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```

<KEY name="showdownloadfolder">
    <VALUE>int</VALUE>
</KEY>
<KEY name="forcedownload">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No



mod_folder_view_folder

Simulate the view.php web interface folder: trigger events, completion, etc...

Arguments

folderid (Required)

folder instance id

General structure

```
int //folder instance id
```

XML-RPC (PHP structure)

```
[folderid] => int
```

REST (POST parameters)

```
folderid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_add_discussion

Add a new discussion into an existing forum.

Arguments**forumid** (Required)

Forum instance ID

General structure

```
int //Forum instance ID
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

subject (Required)

New Discussion subject

General structure

```
string //New Discussion subject
```

XML-RPC (PHP structure)

```
[subject] => string
```

REST (POST parameters)

```
subject= string
```

message (Required)

New Discussion message (only html format allowed)

General structure

```
string //New Discussion message (only html format allowed)
```

XML-RPC (PHP structure)

```
[message] => string
```

REST (POST parameters)

```
message= string
```

groupid (Default to "0")

The group, default to 0

General structure

```
int Default to "0" //The group, default to 0
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

options (Default to "Array ()")

Options

General structure

```
Default to "Array
(
)
" //Options
list of (
object {
name string //The allowed keys (value format) are:
discussionsubscribe (bool); subscribe to the discussion?, default to true
discussionpinned (bool); is the discussion pinned, default to false
inlineattachmentsid (int); the draft file area id for inline attachments
attachmentsid (int); the draft file area id for attachments

value string //The value of the option,
This param is validated in the external function.
}
)
```

XML-RPC (PHP structure)

```
[options] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response

General structure

```

object {
discussionid int //New Discussion ID
warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}

```

XML-RPC (PHP structure)

```

Array
(
[discussionid] => int
[warnings] =>
    Array
    (
[0] =>
        Array
        (
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)

```

REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="discussionid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_forum_add_discussion_post`

Create new posts into an existing discussion.

Arguments

`postid` (Required)

the post id we are going to reply to (can be the initial discussion post)

General structure

```
int //the post id we are going to reply to  
(can be the initial discussion post)
```

XML-RPC (PHP structure)

```
[postid] => int
```

REST (POST parameters)

```
postid= int
```

`subject` (Required)

new post subject

General structure

```
string //new post subject
```

XML-RPC (PHP structure)

```
[subject] => string
```



REST (POST parameters)

```
subject= string
```

`message` (Required)

new post message (html assumed if messageformat is not provided)

General structure

```
string //new post message (html assumed if messageformat is not provided)
```

XML-RPC (PHP structure)

```
[message] => string
```

REST (POST parameters)

```
message= string
```

```
options (Default to "Array ()")
```

Options

General structure

```
Default to "Array"
(
)
" //Options
list of (
object {
name string //The allowed keys (value format) are:
discussionsubscribe (bool); subscribe to the discussion?, default to true
private (bool); make this reply private to the author of the parent post, default to false.
inlineattachmentsid (int); the draft file area id for inline attachments
attachmentsid (int); the draft file area id for attachments
topreferredformat (bool); convert the message & messageformat to FORMAT_HTML, defaults to false

value string //the value of the option,
this param is validated in the external function.
}
)
```

XML-RPC (PHP structure)

```
[options] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

messageformat (Default to "1")

message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN

General structure

```
int Default to "1" //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
```

XML-RPC (PHP structure)

```
[messageformat] => int
```

REST (POST parameters)

```
messageformat= int
```

Response

General structure

```
object {
postid int //new post id
warnings Optional //list of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)post object {
id int //id
subject string //subject
repliesubject string //repliesubject
message string //message
messageformat int //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
author object {
id int Optional //id
fullname string Optional //fullname
isdeleted int Optional //isdeleted
groups Optional //groups
list of (
object {
id int //id
name string //name
urls object {
image string Optional //image
}
)
}
)urls object {
profile string Optional //The URL for the user profile page
profileimage string Optional //The URL for the user profile image
}
)
}
discussionid int //discussionid
hasparent int //hasparent
parentid int Optional //parentid
timecreated int //timecreated
timemodified int //timemodified
unread int Optional //unread
isdeleted int //isdeleted
isprivatereply int //isprivatereply
haswordcount int //haswordcount
wordcount int Optional //wordcount
charcount int Optional //charcount
capabilities object {
view int //Whether the user can view the post
edit int //Whether the user can edit the post
delete int //Whether the user can delete the post
split int //Whether the user can split the post
reply int //Whether the user can reply to the post
selfenrol int //Whether the user can self enrol into the course
export int //Whether the user can export the post
controlreadstatus int //Whether the user can control the read status of the post
canreplyprivately int //Whether the user can post a private reply
}
urls object {
view string Optional //The URL used to view the post
viewisolated string Optional //The URL used to view the post in isolation
viewparent string Optional //The URL used to view the parent of the post
edit string Optional //The URL used to edit the post
delete string Optional //The URL used to delete the post
split string Optional //The URL used to split the discussion with the selected post being the first post in the new discussion
reply string Optional //The URL used to reply to the post
export string Optional //The URL used to export the post
markasread string Optional //The URL used to mark the post as read
markasunread string Optional //The URL used to mark the post as unread
discuss string Optional //discuss
}
attachments //attachments
list of (
object {
contextid int //contextid
component string //component
filearea string //filearea
itemid int //itemid
filepath string //filepath
filename string //filename
isdir int //isdir
isimage int //isimage
}
)
```

```
timemodified int    //timemodified
timecreated int     //timecreated
filesize int        //filesize
author string       //author
license string      //license
filenameshort string //filenameshort
filesizeformatted string //filesizeformatted
icon string         //icon
timecreatedformatted string //timecreatedformatted
timemodifiedformatted string //timemodifiedformatted
url string          //url
urls object {
    export string  Optional //The URL used to export the attachment
}
html object {
    plagiarism string Optional //The HTML source for the Plagiarism Response
}
)
)messageinlinefiles Optional //messageinlinefiles
list of (
object {
    contextid int   //contextid
    component string //component
    filearea string  //filearea
    itemid int      //itemid
    filepath string  //filepath
    filename string  //filename
    isdir int       //isdir
    isimage int     //isimage
    timemodified int //timemodified
    timecreated int  //timecreated
    filesize int    //filesize
    author string   //author
    license string  //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string     //icon
    timecreatedformatted string //timecreatedformatted
    timemodifiedformatted string //timemodifiedformatted
    url string      //url
}
)tags Optional //tags
list of (
object {
    id int          //The ID of the Tag
    tagid int        //The tagid
    isstandard int   //Whether this is a standard tag
    displayname string //The display name of the tag
    flag int         //Whether this tag is flagged
    urls object {
        view string //The URL to view the tag
    }
}
)html object {
    rating string  Optional //The HTML source to rate the post
    taglist string  Optional //The HTML source to view the list of tags
    authorsubheading string Optional //The HTML source to view the author details
}
)
messages Optional //List of warnings
list of (
    //Messages
object {
    type string   //The classification to be used in the client side
    message string //untranslated english message to explain the warning
}
)
)}
```



XML-RPC (PHP structure)

```

Array
(
    [postid] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
    [post] =>
        Array
        (
            [id] => int
            [subject] => string
            [repliesubject] => string
            [message] => string
            [messageformat] => int
            [author] =>
                Array
                (
                    [id] => int
                    [fullname] => string
                    [isdeleted] => int
                    [groups] =>
                        Array
                        (
                            [0] =>
                                Array
                                (
                                    [id] => int
                                    [name] => string
                                    [urls] =>
                                        Array
                                        (
                                            [image] => string
                                        )
                                )
                        )
                )
            [urls] =>
                Array
                (
                    [profile] => string
                    [profileimage] => string
                )
        )
    [discussionid] => int
    [hasparent] => int
    [parentid] => int
    [timecreated] => int
    [timemodified] => int
    [unread] => int
    [isdeleted] => int
    [isprivateresponse] => int
    [haswordcount] => int
    [wordcount] => int
    [charcount] => int
    [capabilities] =>
        Array
        (
            [view] => int
            [edit] => int
            [delete] => int
            [split] => int
            [reply] => int
            [selfenrol] => int
            [export] => int
            [controlreadstatus] => int
            [canreplyprivately] => int
        )
    [urls] =>
        Array
        (
            [view] => string
            [viewisolated] => string
        )
)

```



```
[viewparent] => string
[edit] => string
[delete] => string
[split] => string
[reply] => string
[export] => string
[markasread] => string
[markasunread] => string
[discuss] => string
)
[attachments] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [contextid] => int
                [component] => string
                [filearea] => string
                [itemid] => int
                [filepath] => string
                [filename] => string
                [isdir] => int
                [isimage] => int
                [timemodified] => int
                [timecreated] => int
                [filesize] => int
                [author] => string
                [license] => string
                [filenameshort] => string
                [filesizeformatted] => string
                [icon] => string
                [timecreatedformatted] => string
                [timemodifiedformatted] => string
                [url] => string
                [urls] =>
                  Array
                    (
                      [
                        [export] => string
                      )
                    )
                [html] =>
                  Array
                    (
                      [
                        [plagiarism] => string
                      )
                    )
              )
            )
        )
      )
    )
  )
[messageinlinefiles] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [contextid] => int
                [component] => string
                [filearea] => string
                [itemid] => int
                [filepath] => string
                [filename] => string
                [isdir] => int
                [isimage] => int
                [timemodified] => int
                [timecreated] => int
                [filesize] => int
                [author] => string
                [license] => string
                [filenameshort] => string
                [filesizeformatted] => string
                [icon] => string
                [timecreatedformatted] => string
                [timemodifiedformatted] => string
                [url] => string
              )
            )
        )
      )
    )
  )
[tags] =>
  Array
    (
      [
        [0] =>
          Array
            (
              [
                [id] => int
                [tagid] => int
                [isstandard] => int
              )
            )
        )
      )
    )
  )
)
```



```
[displayname] => string
[flag] => int
[urls] =>
  Array
    (
      [view] => string
    )
  )
)
)
[html] =>
  Array
  (
    [rating] => string
    [taglist] => string
    [authorsubheading] => string
  )
)
)
[messages] =>
  Array
  (
    [0] =>
      Array
      (
        [type] => string
        [message] => string
      )
    )
  )
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="postid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="warnings">
      <MULTIPLE>
        <SINGLE>
          <KEY name="item">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="itemid">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="warningcode">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="post">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="subject">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="repliesubject">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="message">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="messageformat">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="author">
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdeleted">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="groups">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="id">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="name">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="urls">
                    <SINGLE>
                      <KEY name="image">
                        <VALUE>string</VALUE>
                      </KEY>
                    </SINGLE>
                  </KEY>
                </SINGLE>
              </MULTIPLE>
            </KEY>
            <KEY name="urls">
              <SINGLE>
                <KEY name="profile">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="profileimage">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</RESPONSE>
```

```
</SINGLE>
</KEY>
<KEY name="discussionid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasparent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="unread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isdeleted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isprivatereply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="haswordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="wordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="charcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="capabilities">
    <SINGLE>
        <KEY name="view">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="selfenrol">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="controlreadstatus">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplyprivately">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewisolated">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewparent">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```
<KEY name="split">
    <VALUE>string</VALUE>
</KEY>
<KEY name="reply">
    <VALUE>string</VALUE>
</KEY>
<KEY name="export">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasunread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="discuss">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="attachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodifiedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="urls">
                <SINGLE>
                    <KEY name="export">
```



```
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="html">
<SINGLE>
<KEY name="plagiarism">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messageinlinefiles">
<MULTIPLE>
<SINGLE>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="component">
<VALUE>string</VALUE>
</KEY>
<KEY name="filearea">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="filepath">
<VALUE>string</VALUE>
</KEY>
<KEY name="filename">
<VALUE>string</VALUE>
</KEY>
<KEY name="isdir">
<VALUE>int</VALUE>
</KEY>
<KEY name="isimage">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="filesize">
<VALUE>int</VALUE>
</KEY>
<KEY name="author">
<VALUE>string</VALUE>
</KEY>
<KEY name="license">
<VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
<VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="icon">
<VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="url">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
```



```

<KEY name="tagid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isstandard">
    <VALUE>int</VALUE>
</KEY>
<KEY name="displayname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="flag">
    <VALUE>int</VALUE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="html">
    <SINGLE>
        <KEY name="rating">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="taglist">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="authorsubheading">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<SINGLE>
</MULTIPLE>
<KEY name="messages">
    <MULTIPLE>
        <SINGLE>
            <KEY name="type">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

[mod_forum_can_add_discussion](#)

Check if the current user can add discussions in the given forum (and optionally for the given group).

Arguments

forumid (Required)

Forum instance ID

General structure

```
int //Forum instance ID
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

groupid (Default to "null")

The group to check, default to active group. Use -1 to check if the user can post in all the groups.

General structure

```
int Default to "null" //The group to check, default to active group.  
Use -1 to check if the user can post in all the groups.
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

Response

General structure

```
object {  
status int //True if the user can add discussions, false otherwise.  
canpindiscussions int Optional //True if the user can pin discussions, false otherwise.  
cancreateattachment int Optional //True if the user can add attachments, false otherwise.  
warnings Optional //List of warnings  
list of (  
//warning  
object {  
item string Optional //item  
itemid int Optional //item id  
warningcode string //the warning code can be used by the client app to implement specific behaviour  
message string //untranslated english message to explain the warning  
}  
)}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [canpindiscussions] => int
    [cancreateattachment] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpindiscussions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cancreateattachment">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_delete_post

Deletes a post or a discussion completely when the post is the discussion topic.

Arguments**postid** (Required)

Post to be deleted. It can be a discussion topic post.

General structure

```
int //Post to be deleted. It can be a discussion topic post.
```

XML-RPC (PHP structure)

```
[postid] => int
```

REST (POST parameters)

```
postid= int
```

Response**General structure**

```
object {
    status int //True if the post/discussion was deleted, false otherwise.
    warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_forum_get_discussion_post**

Get a particular discussion post.

Arguments**postid** (Required)

Post to fetch.

General structure

```
int //Post to fetch.
```

XML-RPC (PHP structure)

```
[postid] => int
```

```
REST (POST parameters)
```

```
postid= int
```

Response

^

```
General structure

object {
post object {
id int //id
subject string //subject
repliesubject string //repliesubject
message string //message
messageformat int //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
author object {
id int Optional //id
fullname string Optional //fullname
isdeleted int Optional //isdeleted
groups Optional //groups
list of (
object {
id int //id
name string //name
urls object {
image string Optional //image
}
)
)
urls object {
profile string Optional //The URL for the user profile page
profileimage string Optional //The URL for the user profile image
}
)
discussionid int //discussionid
hasparent int //hasparent
parentid int Optional //parentid
timecreated int //timecreated
timemodified int //timemodified
unread int Optional //unread
isdeleted int //isdeleted
isprivatereply int //isprivatereply
haswordcount int //haswordcount
wordcount int Optional //wordcount
charcount int Optional //charcount
capabilities object {
view int //Whether the user can view the post
edit int //Whether the user can edit the post
delete int //Whether the user can delete the post
split int //Whether the user can split the post
reply int //Whether the user can reply to the post
selfenrol int //Whether the user can self enrol into the course
export int //Whether the user can export the post
controlreadstatus int //Whether the user can control the read status of the post
canreplyprivately int //Whether the user can post a private reply
}
urls object {
view string Optional //The URL used to view the post
viewisolated string Optional //The URL used to view the post in isolation
viewparent string Optional //The URL used to view the parent of the post
edit string Optional //The URL used to edit the post
delete string Optional //The URL used to delete the post
split string Optional //The URL used to split the discussion with the selected post being the first post in new discussion
reply string Optional //The URL used to reply to the post
export string Optional //The URL used to export the post
markasread string Optional //The URL used to mark the post as read
markasunread string Optional //The URL used to mark the post as unread
discuss string Optional //discuss
}
attachments //attachments
list of (
object {
contextid int //contextid
component string //component
filearea string //filearea
itemid int //itemid
filepath string //filepath
filename string //filename
isdir int //isdir
isimage int //isimage
timemodified int //timemodified
timecreated int //timecreated
filesize int //filesize
author string //author
license string //license
filenameshort string //filenameshort
filesizeformatted string //filesizeformatted
icon string //icon
timecreatedformatted string //timecreatedformatted
timemodifiedformatted string //timemodifiedformatted
}
```

```
url string //url
urls object {
    export string Optional //The URL used to export the attachment
}
html object {
    plagiarism string Optional //The HTML source for the Plagiarism Response
}
)
)messageinlinefiles Optional //messageinlinefiles
list of (
object {
    contextid int //contextid
    component string //component
    filearea string //filearea
    itemid int //itemid
    filepath string //filepath
    filename string //filename
    isdir int //isdir
    isimage int //isimage
    timemodified int //timemodified
    timecreated int //timecreated
    filesize int //filesize
    author string //author
    license string //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string //icon
    timecreatedformatted string //timecreatedformatted
    timemodifiedformatted string //timemodifiedformatted
    url string //url
}
)tags Optional //tags
list of (
object {
    id int //The ID of the Tag
    tagid int //The tagid
    isstandard int //Whether this is a standard tag
    displayname string //The display name of the tag
    flag int //Whether this tag is flagged
    urls object {
        view string //The URL to view the tag
    }
}
)
)html object {
    rating string Optional //The HTML source to rate the post
    taglist string Optional //The HTML source to view the list of tags
    authorsubheading string Optional //The HTML source to view the author details
}
)
)
warnings Optional //List of warnings
list of (
    //warning
object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
}
)
)}
```



XML-RPC (PHP structure)

```

Array
(
    [post] =>
        Array
            (
                [id] => int
                [subject] => string
                [repliesubject] => string
                [message] => string
                [messageformat] => int
                [author] =>
                    Array
                        (
                            [id] => int
                            [fullname] => string
                            [isdeleted] => int
                            [groups] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [id] => int
                                                    [name] => string
                                                    [urls] =>
                                                        Array
                                                            (
                                                                [image] => string
                                                            )
                                                )
                                        )
                                    )
                            [urls] =>
                                Array
                                    (
                                        [profile] => string
                                        [profileimage] => string
                                    )
                        )
                [discussionid] => int
                [hasparent] => int
                [parentid] => int
                [timecreated] => int
                [timemodified] => int
                [unread] => int
                [isdeleted] => int
                [isprivateresponse] => int
                [haswordcount] => int
                [wordcount] => int
                [charcount] => int
                [capabilities] =>
                    Array
                        (
                            [view] => int
                            [edit] => int
                            [delete] => int
                            [split] => int
                            [reply] => int
                            [selfenrol] => int
                            [export] => int
                            [controlreadstatus] => int
                            [canreplyprivately] => int
                        )
                [urls] =>
                    Array
                        (
                            [view] => string
                            [viewisolated] => string
                            [viewparent] => string
                            [edit] => string
                            [delete] => string
                            [split] => string
                            [reply] => string
                            [export] => string
                            [markasread] => string
                            [markasunread] => string
                            [discuss] => string
                        )
            )
        [attachments] =>
            Array
                (

```



```
[0] =>
  Array
    (
      [
        [contextid] => int
        [component] => string
        [filearea] => string
        [itemid] => int
        [filepath] => string
        [filename] => string
        [isdir] => int
        [isimage] => int
        [timemodified] => int
        [timecreated] => int
        [filesize] => int
        [author] => string
        [license] => string
        [filenameshort] => string
        [filesizeformatted] => string
        [icon] => string
        [timecreatedformatted] => string
        [timemodifiedformatted] => string
        [url] => string
        [urls] =>
          Array
            (
              [
                [export] => string
              ]
            )
      [
        [html] =>
          Array
            (
              [
                [plagiarism] => string
              ]
            )
      ]
    )
  [
    [messageinlinefiles] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [contextid] => int
                    [component] => string
                    [filearea] => string
                    [itemid] => int
                    [filepath] => string
                    [filename] => string
                    [isdir] => int
                    [isimage] => int
                    [timemodified] => int
                    [timecreated] => int
                    [filesize] => int
                    [author] => string
                    [license] => string
                    [filenameshort] => string
                    [filesizeformatted] => string
                    [icon] => string
                    [timecreatedformatted] => string
                    [timemodifiedformatted] => string
                    [url] => string
                  ]
                )
          ]
        )
  [
    [tags] =>
      Array
        (
          [
            [0] =>
              Array
                (
                  [
                    [id] => int
                    [tagid] => int
                    [isstandard] => int
                    [displayname] => string
                    [flag] => int
                    [urls] =>
                      Array
                        (
                          [
                            [view] => string
                          ]
                        )
                  ]
                )
          ]
        )
  [
    [html] =>
      Array
        (
          [
            [rating] => string
          ]
        )
  ]
]
```



```
[taglist] => string
[authorsubheading] => string
)
)
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="post">
      <SINGLE>
        <KEY name="id">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="subject">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="replysubject">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="message">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="messageformat">
          <VALUE>int</VALUE>
        </KEY>
        <KEY name="author">
          <SINGLE>
            <KEY name="id">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="fullname">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdeleted">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="groups">
              <MULTIPLE>
                <SINGLE>
                  <KEY name="id">
                    <VALUE>int</VALUE>
                  </KEY>
                  <KEY name="name">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="urls">
                    <SINGLE>
                      <KEY name="image">
                        <VALUE>string</VALUE>
                      </KEY>
                    </SINGLE>
                  </KEY>
                </SINGLE>
              </MULTIPLE>
            </KEY>
            <KEY name="urls">
              <SINGLE>
                <KEY name="profile">
                  <VALUE>string</VALUE>
                </KEY>
                <KEY name="profileimage">
                  <VALUE>string</VALUE>
                </KEY>
              </SINGLE>
            </KEY>
            <KEY name="discussionid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="hasparent">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="parentid">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="unread">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="isdeleted">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="isprivatereply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="haswordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="wordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="charcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="capabilities">
    <SINGLE>
        <KEY name="view">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="selfenrol">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="controlreadstatus">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplyprivately">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewisolated">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewparent">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasunread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="discuss">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="attachments">
```



```
<MULTIPLE>
  <SINGLE>
    <KEY name="contextid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="component">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="filearea">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="itemid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="filepath">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="filename">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="isdir">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="isimage">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="filesize">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="author">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="license">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="filenameshort">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="filesizeformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="icon">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="timecreatedformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="timemodifiedformatted">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="url">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="urls">
      <SINGLE>
        <KEY name="export">
          <VALUE>string</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
    <KEY name="html">
      <SINGLE>
        <KEY name="plagiarism">
          <VALUE>string</VALUE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</MULTIPLE>
<KEY name="messageinlinefiles">
  <MULTIPLE>
    <SINGLE>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="component">
```

```
<VALUE>string</VALUE>
</KEY>
<KEY name="filearea">
    <VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filepath">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filename">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isdir">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isimage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filesize">
    <VALUE>int</VALUE>
</KEY>
<KEY name="author">
    <VALUE>string</VALUE>
</KEY>
<KEY name="license">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isstandard">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="displayname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="flag">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="urls">
                <SINGLE>
                    <KEY name="view">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

        </KEY>
        <KEY name="html">
            <SINGLE>
                <KEY name="rating">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="taglist">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="authorsubheading">
                    <VALUE>string</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_get_discussion_posts

Returns a list of forum posts for a discussion.

Arguments

discussionid (Required)

The ID of the discussion from which to fetch posts.

General structure

```
int //The ID of the discussion from which to fetch posts.
```

XML-RPC (PHP structure)

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

sortby (Default to "created")

Sort by this element: id, created or modified

General structure

```
string Default to "created" //Sort by this element: id, created or modified
```

XML-RPC (PHP structure)

```
[sortby] => string
```

REST (POST parameters)

```
sortby= string
```

sortdirection (Default to "DESC")

Sort direction: ASC or DESC

General structure

```
string Default to "DESC" //Sort direction: ASC or DESC
```

XML-RPC (PHP structure)

```
[sortdirection] => string
```

REST (POST parameters)

```
sortdirection= string
```



includeinlineattachments (Default to "")

Whether inline attachments should be included or not

General structure

```
int Default to "" //Whether inline attachments should be included or not
```

XML-RPC (PHP structure)

```
[includeinlineattachments] => int
```

REST (POST parameters)

```
includeinlineattachments= int
```

Response

^


```
timemodifiedformatted string //timemodifiedformatted
url string //url
urls object {
    export string Optional //The URL used to export the attachment
}
html object {
    plagiarism string Optional //The HTML source for the Plagiarism Response
}
)
)messageinlinefiles Optional //messageinlinefiles
list of (
object {
    contextid int //contextid
    component string //component
    filearea string //filearea
    itemid int //itemid
    filepath string //filepath
    filename string //filename
    isdir int //isdir
    isimage int //isimage
    timemodified int //timemodified
    timecreated int //timecreated
    filesize int //filesize
    author string //author
    license string //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string //icon
    timecreatedformatted string //timecreatedformatted
    timemodifiedformatted string //timemodifiedformatted
    url string //url
)
)
)tags Optional //tags
list of (
object {
    id int //The ID of the Tag
    tagid int //The tagid
    isstandard int //Whether this is a standard tag
    displayname string //The display name of the tag
    flag int //Wehther this tag is flagged
    urls object {
        view string //The URL to view the tag
    }
)
)
)html object {
    rating string Optional //The HTML source to rate the post
    taglist string Optional //The HTML source to view the list of tags
    authortsubheading string Optional //The HTML source to view the author details
)
)
)
)forumid int //The forum id
courseid int //The forum course id
ratinginfo Optional //Rating information
object {
    contextid int //Context id.
    component string //Context name.
    ratingarea string //Rating area name.
    canviewall int Optional //Whether the user can view all the individual ratings.
    canviewany int Optional //Whether the user can view aggregate of ratings of others.
    scales Optional //Different scales used information
    list of (
        //Scale information
        object {
            id int //Scale id.
            courseid int Optional //Course id.
            name string Optional //Scale name (when a real scale is used).
            max int //Max value for the scale.
            isnumeric int //Whether is a numeric scale.
            items Optional //Scale items. Only returned for not numerical scales.
            list of (
                object {
                    value int //Scale value/option id.
                    name string //Scale name.
                }
            )
        }
    )
)
)ratings Optional //The ratings
list of (
object {
    itemid int //Item id.
    scaleid int Optional //Scale id.
    userid int Optional //User who rated id.
    aggregate double Optional //Aggregated ratings grade.
    aggregatestr string Optional //Aggregated ratings as string.
}
```

```
aggregateLabel string Optional //The aggregation label.  
count int Optional //Ratings count (used when aggregating).  
rating int Optional //The rating the user gave.  
canRate int Optional //Whether the user can rate the item.  
canViewAggregate int Optional //Whether the user can view the aggregated grade.  
}  
}  
}  
}  
warnings Optional //List of warnings  
list of (  
    //warning  
object {  
item string Optional //item  
itemId int Optional //item id  
warningCode string //the warning code can be used by the client app to implement specific behaviour  
message string //untranslated english message to explain the warning  
}  
}  
}
```



XML-RPC (PHP structure)

```

Array
(
[posts] =>
    Array
        (
            [0] =>
                Array
                    (
                        [id] => int
                        [subject] => string
                        [repliesubject] => string
                        [message] => string
                        [messageformat] => int
                        [author] =>
                            Array
                                (
                                    [id] => int
                                    [fullname] => string
                                    [isdeleted] => int
                                    [groups] =>
                                        Array
                                            (
                                                [0] =>
                                                    Array
                                                        (
                                                            [id] => int
                                                            [name] => string
                                                            [urls] =>
                                                                Array
                                                                    (
                                                                        [image] => string
                                                                    )
                                                        )
                                                )
                                            )
                                    [urls] =>
                                        Array
                                            (
                                                [profile] => string
                                                [profileimage] => string
                                            )
                                        )
                                    )
                                )
                            )
                        [discussionid] => int
                        [hasparent] => int
                        [parentid] => int
                        [timecreated] => int
                        [timemodified] => int
                        [unread] => int
                        [isdeleted] => int
                        [isprivatereply] => int
                        [haswordcount] => int
                        [wordcount] => int
                        [charcount] => int
                        [capabilities] =>
                            Array
                                (
                                    [view] => int
                                    [edit] => int
                                    [delete] => int
                                    [split] => int
                                    [reply] => int
                                    [selfenrol] => int
                                    [export] => int
                                    [controlreadstatus] => int
                                    [canreplyprivately] => int
                                )
                        [urls] =>
                            Array
                                (
                                    [view] => string
                                    [viewisolated] => string
                                    [viewparent] => string
                                    [edit] => string
                                    [delete] => string
                                    [split] => string
                                    [reply] => string
                                    [export] => string
                                    [markasread] => string
                                    [markasunread] => string
                                    [discuss] => string
                                )
                    )
                )
            )
        )
)

```

^

```
[attachments] =>
  Array
    (
      [0] =>
        Array
          (
            [contextid] => int
            [component] => string
            [filearea] => string
            [itemid] => int
            [filepath] => string
            [filename] => string
            [isdir] => int
            [isimage] => int
            [timemodified] => int
            [timecreated] => int
            [filesize] => int
            [author] => string
            [license] => string
            [filenameshort] => string
            [filesizeformatted] => string
            [icon] => string
            [timecreatedformatted] => string
            [timemodifiedformatted] => string
            [url] => string
            [urls] =>
              Array
                (
                  [export] => string
                )
            [html] =>
              Array
                (
                  [plagiarism] => string
                )
          )
      )
    )
[messageinlinefiles] =>
  Array
    (
      [0] =>
        Array
          (
            [contextid] => int
            [component] => string
            [filearea] => string
            [itemid] => int
            [filepath] => string
            [filename] => string
            [isdir] => int
            [isimage] => int
            [timemodified] => int
            [timecreated] => int
            [filesize] => int
            [author] => string
            [license] => string
            [filenameshort] => string
            [filesizeformatted] => string
            [icon] => string
            [timecreatedformatted] => string
            [timemodifiedformatted] => string
            [url] => string
          )
    )
  )
[tags] =>
  Array
    (
      [0] =>
        Array
          (
            [id] => int
            [tagid] => int
            [isstandard] => int
            [displayname] => string
            [flag] => int
            [urls] =>
              Array
                (
                  [view] => string
                )
          )
    )
  )
[html] =>
```



```
        Array
        (
            [rating] => string
            [taglist] => string
            [authorsubheading] => string
        )
    )
)
[forumid] => int
[courseid] => int
[ratinginfo] =>
    Array
    (
        [contextid] => int
        [component] => string
        [ratingarea] => string
        [canviewall] => int
        [canviewany] => int
        [scales] =>
            Array
            (
                [0] =>
                    Array
                    (
                        [id] => int
                        [courseid] => int
                        [name] => string
                        [max] => int
                        [isnumeric] => int
                        [items] =>
                            Array
                            (
                                [0] =>
                                    Array
                                    (
                                        [value] => int
                                        [name] => string
                                    )
                            )
                    )
            )
        )
    )
[ratings] =>
    Array
    (
        [0] =>
            Array
            (
                [itemid] => int
                [scaleid] => int
                [userid] => int
                [aggregate] => double
                [aggregatestr] => string
                [aggregatelabel] => string
                [count] => int
                [rating] => int
                [canrate] => int
                [canviewaggregate] => int
            )
    )
)
)
[warnings] =>
    Array
    (
        [0] =>
            Array
            (
                [item] => string
                [itemid] => int
                [warningcode] => string
                [message] => string
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="posts">
      <MULTIPLE>
        <SINGLE>
          <KEY name="id">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="subject">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="repliesubject">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="message">
            <VALUE>string</VALUE>
          </KEY>
          <KEY name="messageformat">
            <VALUE>int</VALUE>
          </KEY>
          <KEY name="author">
            <SINGLE>
              <KEY name="id">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="fullname">
                <VALUE>string</VALUE>
              </KEY>
              <KEY name="isdeleted">
                <VALUE>int</VALUE>
              </KEY>
              <KEY name="groups">
                <MULTIPLE>
                  <SINGLE>
                    <KEY name="id">
                      <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                      <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="urls">
                      <SINGLE>
                        <KEY name="image">
                          <VALUE>string</VALUE>
                        </KEY>
                      </SINGLE>
                    </KEY>
                  </SINGLE>
                </MULTIPLE>
              </KEY>
              <KEY name="urls">
                <SINGLE>
                  <KEY name="profile">
                    <VALUE>string</VALUE>
                  </KEY>
                  <KEY name="profileimage">
                    <VALUE>string</VALUE>
                  </KEY>
                </SINGLE>
              </KEY>
            </SINGLE>
          </KEY>
        </SINGLE>
      </MULTIPLE>
    </KEY>
    <KEY name="discussionid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="hasparent">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="parentid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timecreated">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="timemodified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="unread">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</RESPONSE>
```



```
<KEY name="isdeleted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isprivateresponse">
    <VALUE>int</VALUE>
</KEY>
<KEY name="haswordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="wordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="charcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="capabilities">
    <SINGLE>
        <KEY name="view">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="selfenrol">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="controlreadstatus">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplyprivately">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewisolated">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewparent">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasunread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="discuss">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```
<KEY name="attachments">
  <MULTIPLE>
    <SINGLE>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="component">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="filearea">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="filepath">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="filename">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="isdir">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="isimage">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timemodified">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="timecreated">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="filesize">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="author">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="license">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="filenameshort">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="filesizeformatted">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="icon">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timecreatedformatted">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="timemodifiedformatted">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="url">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="urls">
        <SINGLE>
          <KEY name="export">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </KEY>
      <KEY name="html">
        <SINGLE>
          <KEY name="plagiarism">
            <VALUE>string</VALUE>
          </KEY>
        </SINGLE>
      </KEY>
      <SINGLE>
    </SINGLE>
  </MULTIPLE>
</KEY>
<KEY name="messageinlinefiles">
  <MULTIPLE>
    <SINGLE>
      <KEY name="contextid">
        <VALUE>int</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</KEY>
```



```
<KEY name="component">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filearea">
    <VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filepath">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filename">
    <VALUE>string</VALUE>
</KEY>
<KEY name="isdir">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isimage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="filesize">
    <VALUE>int</VALUE>
</KEY>
<KEY name="author">
    <VALUE>string</VALUE>
</KEY>
<KEY name="license">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
    <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isstandard">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="displayname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="flag">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="urls">
                <SINGLE>
                    <KEY name="view">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```
</MULTIPLE>
</KEY>
<KEY name="html1">
<SINGLE>
<KEY name="rating">
<VALUE>string</VALUE>
</KEY>
<KEY name="taglist">
<VALUE>string</VALUE>
</KEY>
<KEY name="authorsubheading">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="forumid">
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
<VALUE>int</VALUE>
</KEY>
<KEY name="ratinginfo">
<SINGLE>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="component">
<VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
<VALUE>string</VALUE>
</KEY>
<KEY name="canviewall">
<VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
<VALUE>int</VALUE>
</KEY>
<KEY name="scales">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="max">
<VALUE>int</VALUE>
</KEY>
<KEY name="isnumeric">
<VALUE>int</VALUE>
</KEY>
<KEY name="items">
<MULTIPLE>
<SINGLE>
<KEY name="value">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratings">
<MULTIPLE>
<SINGLE>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
<VALUE>int</VALUE>
</KEY>
```



```

<KEY name="userid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="aggregate">
    <VALUE>double</VALUE>
</KEY>
<KEY name="aggregatestr">
    <VALUE>string</VALUE>
</KEY>
<KEY name="aggregatelabel">
    <VALUE>string</VALUE>
</KEY>
<KEY name="count">
    <VALUE>int</VALUE>
</KEY>
<KEY name="rating">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canrate">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canviewaggregate">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="warningcode">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message



REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_forum_get_discussion_posts_by_userid

Returns a list of forum posts for a discussion for a user.

Arguments

userid (Required)

The ID of the user of whom to fetch posts.

General structure

```
int //The ID of the user of whom to fetch posts.
```

XML-RPC (PHP structure)

```
[userid] => int
```

REST (POST parameters)

```
userid= int
```

cmid (Required)

The ID of the module of which to fetch items.

General structure

```
int //The ID of the module of which to fetch items.
```

XML-RPC (PHP structure)

```
[cmid] => int
```

REST (POST parameters)

```
cmid= int
```

sortby (Default to "created")

Sort by this element: id, created or modified

General structure

```
string Default to "created" //Sort by this element: id, created or modified
```

XML-RPC (PHP structure)

```
[sortby] => string
```

REST (POST parameters)

```
sortby= string
```

sortdirection (Default to "DESC")

Sort direction: ASC or DESC

General structure

```
string Default to "DESC" //Sort direction: ASC or DESC
```

XML-RPC (PHP structure)

```
[sortdirection] => string
```

```
REST (POST parameters)
```

```
sortdirection= string
```

Response

^

```
General structure

object {
discussions list of (
object {
name string //Name of the discussion
id int //ID of the discussion
timecreated int //Timestamp of the discussion start
authorfullname string //Full name of the user that started the discussion
posts object {
userposts list of (
object {
id int //id
subject string //subject
repliesubject string //repliesubject
message string //message
messageformat int //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
author object {
id int Optional //id
fullname string Optional //fullname
isdeleted int Optional //isdeleted
groups Optional //groups
list of (
object {
id int //id
name string //name
urls object {
image string Optional //image
}
}
)
)urls object {
profile string Optional //The URL for the user profile page
profileimage string Optional //The URL for the user profile image
}
}
discussionid int //discussionid
hasparent int //hasparent
parentid int Optional //parentid
timecreated int //timecreated
timemodified int //timemodified
unread int Optional //unread
isdeleted int //isdeleted
isprivateresponse int //isprivateresponse
haswordcount int //haswordcount
wordcount int Optional //wordcount
charcount int Optional //charcount
capabilities object {
view int //Whether the user can view the post
edit int //Whether the user can edit the post
delete int //Whether the user can delete the post
split int //Whether the user can split the post
reply int //Whether the user can reply to the post
selfenrol int //Whether the user can self enrol into the course
export int //Whether the user can export the post
controlreadstatus int //Whether the user can control the read status of the post
canreplyprivately int //Whether the user can post a private reply
}
urls object {
view string Optional //The URL used to view the post
viewisolated string Optional //The URL used to view the post in isolation
viewparent string Optional //The URL used to view the parent of the post
edit string Optional //The URL used to edit the post
delete string Optional //The URL used to delete the post
split string Optional //The URL used to split the discussion with the selected post being the first post in the new discussion
reply string Optional //The URL used to reply to the post
export string Optional //The URL used to export the post
markasread string Optional //The URL used to mark the post as read
markasunread string Optional //The URL used to mark the post as unread
discuss string Optional //discuss
}
attachments //attachments
list of (
object {
contextid int //contextid
component string //component
filearea string //filearea
itemid int //itemid
filepath string //filepath
filename string //filename
isdir int //isdir
isimage int //isimage
timemodified int //timemodified
timecreated int //timecreated
}
)
}
```

```
filesize int //filesize
author string //author
license string //license
filenameshort string //filenameshort
filesizeformatted string //filesizeformatted
icon string //icon
timecreatedformatted string //timecreatedformatted
timemodifiedformatted string //timemodifiedformatted
url string //url
urls object {
    export string Optional //The URL used to export the attachment
}
html object {
    plagiarism string Optional //The HTML source for the Plagiarism Response
}
)
)messageinlinefiles Optional //messageinlinefiles
list of (
object {
    contextid int //contextid
    component string //component
    filearea string //filearea
    itemid int //itemid
    filepath string //filepath
    filename string //filename
    isdir int //isdir
    isimage int //isimage
    timemodified int //timemodified
    timecreated int //timecreated
    filesize int //filesize
    author string //author
    license string //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string //icon
    timecreatedformatted string //timecreatedformatted
    timemodifiedformatted string //timemodifiedformatted
    url string //url
}
)tags Optional //tags
list of (
object {
    id int //The ID of the Tag
    tagid int //The tagid
    isstandard int //Whether this is a standard tag
    displayname string //The display name of the tag
    flag int //Whether this tag is flagged
    urls object {
        view string //The URL to view the tag
    }
}
)html object {
    rating string Optional //The HTML source to rate the post
    taglist string Optional //The HTML source to view the List of tags
    authortitle string Optional //The HTML source to view the author details
}
)
)parenposts list of (
object {
    id int //id
    subject string //subject
    replysubject string //replysubject
    message string //message
    messageformat int //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    author object {
        id int Optional //id
        fullname string Optional //fullname
        isdeleted int Optional //isdeleted
        groups Optional //groups
        list of (
        object {
            id int //id
            name string //name
            urls object {
                image string Optional //image
            }
        }
        )urls object {
            profile string Optional //The URL for the user profile page
            profileimage string Optional //The URL for the user profile image
        }
    }
    discussionid int //discussionid
}
```



```

hasparent int //hasparent
parentid int Optional //parentid
timecreated int //timecreated
timemodified int //timemodified
unread int Optional //unread
isdeleted int //isdeleted
isprivatereply int //isprivatereply
haswordcount int //haswordcount
wordcount int Optional //wordcount
charcount int Optional //charcount
capabilities object {
    view int //Whether the user can view the post
    edit int //Whether the user can edit the post
    delete int //Whether the user can delete the post
    split int //Whether the user can split the post
    reply int //Whether the user can reply to the post
    selfenrol int //Whether the user can self enrol into the course
    export int //Whether the user can export the post
    controlreadstatus int //Whether the user can control the read status of the post
    canreplyprivately int //Whether the user can post a private reply
}
urls object {
    view string Optional //The URL used to view the post
    viewisolated string Optional //The URL used to view the post in isolation
    viewparent string Optional //The URL used to view the parent of the post
    edit string Optional //The URL used to edit the post
    delete string Optional //The URL used to delete the post
    split string Optional //The URL used to split the discussion with the selected post being the first post in the new discussion
    reply string Optional //The URL used to reply to the post
    export string Optional //The URL used to export the post
    markasread string Optional //The URL used to mark the post as read
    markasunread string Optional //The URL used to mark the post as unread
    discuss string Optional //discuss
}
attachments //attachments
list of (
object {
    contextid int //contextid
    component string //component
    filearea string //filearea
    itemid int //itemid
    filepath string //filepath
    filename string //filename
    isdir int //isdir
    isimage int //isimage
    timemodified int //timemodified
    timecreated int //timecreated
    filesize int //filesize
    author string //author
    license string //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string //icon
    timecreatedformatted string //timecreatedformatted
    timemodifiedformatted string //timemodifiedformatted
    url string //url
}
urls object {
    export string Optional //The URL used to export the attachment
}
html object {
    plagiarism string Optional //The HTML source for the Plagiarism Response
}
)
messageinlinefiles Optional //messageinlinefiles
list of (
object {
    contextid int //contextid
    component string //component
    filearea string //filearea
    itemid int //itemid
    filepath string //filepath
    filename string //filename
    isdir int //isdir
    isimage int //isimage
    timemodified int //timemodified
    timecreated int //timecreated
    filesize int //filesize
    author string //author
    license string //license
    filenameshort string //filenameshort
    filesizeformatted string //filesizeformatted
    icon string //icon
    timecreatedformatted string //timecreatedformatted
}
)

```

```
timemodifiedformatted string //timemodifiedformatted
url string //url
}
)tags Optional //tags
list of (
object {
id int //The ID of the Tag
tagid int //The tagid
isstandard int //Whether this is a standard tag
displayname string //The display name of the tag
flag int //Whether this tag is flagged
urls object {
view string //The URL to view the tag
}
}
)
)html object {
rating string Optional //The HTML source to rate the post
taglist string Optional //The HTML source to view the List of tags
authorsubheading string Optional //The HTML source to view the author details
}
}
)
)
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
```



XML-RPC (PHP structure)

```

Array
(
[discussions] =>
    Array
    (
        [0] =>
            Array
            (
                [name] => string
                [id] => int
                [timecreated] => int
                [authorfullname] => string
                [posts] =>
                    Array
                    (
                        [0] =>
                            Array
                            (
                                [userposts] =>
                                    Array
                                    (
                                        [0] =>
                                            Array
                                            (
                                                [id] => int
                                                [subject] => string
                                                [repliesubject] => string
                                                [message] => string
                                                [messageformat] => int
                                                [author] =>
                                                    Array
                                                    (
                                                        [id] => int
                                                        [fullname] => string
                                                        [isdeleted] => int
                                                        [groups] =>
                                                            Array
                                                            (
                                                                [0] =>
                                                                    Array
                                                                    (
                                                                        [id] => int
                                                                        [name] => string
                                                                        [urls] =>
                                                                            Array
                                                                            (
                                                                                [image] => string
                                                                            )
                                                                    )
                                                                )
                                                        )
                                                    )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
[discussionid] => int
[hasparent] => int
[parentid] => int
[timecreated] => int
[timemodified] => int
[unread] => int
[isdeleted] => int
[isprivatereply] => int
[haswordcount] => int
[wordcount] => int
[charcount] => int
[capabilities] =>
    Array
    (
        [view] => int
        [edit] => int
        [delete] => int
        [split] => int
        [reply] => int
        [selfenrol] => int
        [export] => int
        [controlreadstatus] => int
        [canreplyprivately] => int
    )
)
[urls] =>
    Array

```

```
(  
  [view] => string  
  [viewisolated] => string  
  [viewparent] => string  
  [edit] => string  
  [delete] => string  
  [split] => string  
  [reply] => string  
  [export] => string  
  [markasread] => string  
  [markasunread] => string  
  [discuss] => string  
  )  
)  
[attachments] =>  
  Array  
  (  
    [0] =>  
      Array  
      (  
        [contextid] => int  
        [component] => string  
        [filearea] => string  
        [itemid] => int  
        [filepath] => string  
        [filename] => string  
        [isdir] => int  
        [isimage] => int  
        [timemodified] => int  
        [timecreated] => int  
        [filesize] => int  
        [author] => string  
        [license] => string  
        [filenameshort] => string  
        [filesizeformatted] => string  
        [icon] => string  
        [timecreatedformatted] => string  
        [timemodifiedformatted] => string  
        [url] => string  
        [urls] =>  
          Array  
          (  
            [export] => string  
          )  
        )  
      [html] =>  
        Array  
        (  
          [plagiarism] => string  
        )  
      )  
    )  
  )  
)[messageinlinefiles] =>  
  Array  
  (  
    [0] =>  
      Array  
      (  
        [contextid] => int  
        [component] => string  
        [filearea] => string  
        [itemid] => int  
        [filepath] => string  
        [filename] => string  
        [isdir] => int  
        [isimage] => int  
        [timemodified] => int  
        [timecreated] => int  
        [filesize] => int  
        [author] => string  
        [license] => string  
        [filenameshort] => string  
        [filesizeformatted] => string  
        [icon] => string  
        [timecreatedformatted] => string  
        [timemodifiedformatted] => string  
        [url] => string  
      )  
    )  
  )[tags] =>  
  Array  
  (  
    [0] =>  
      Array  
      (  
        ^
```



```
        )
[urls] =>
  Array
    (
      [view] => string
      [viewisolated] => string
      [viewparent] => string
      [edit] => string
      [delete] => string
      [split] => string
      [reply] => string
      [export] => string
      [markasread] => string
      [markasunread] => string
      [discuss] => string
    )
[attachments] =>
  Array
    (
      [0] =>
        Array
          (
            [
              [contextid] => int
              [component] => string
              [filearea] => string
              [itemid] => int
              [filepath] => string
              [filename] => string
              [isdir] => int
              [isimage] => int
              [timemodified] => int
              [timecreated] => int
              [filesize] => int
              [author] => string
              [license] => string
              [filenameshort] => string
              [filesizeformatted] => string
              [icon] => string
              [timecreatedformatted] => string
              [timemodifiedformatted] => string
              [url] => string
            ]
            [
              [urls] =>
                Array
                  (
                    [
                      [
                        [export] => string
                      ]
                    ]
                  )
            ]
            [
              [html] =>
                Array
                  (
                    [
                      [
                        [plagiarism] => string
                      ]
                    ]
                  )
            ]
          )
        )
      )
[messageinlinefiles] =>
  Array
    (
      [0] =>
        Array
          (
            [
              [contextid] => int
              [component] => string
              [filearea] => string
              [itemid] => int
              [filepath] => string
              [filename] => string
              [isdir] => int
              [isimage] => int
              [timemodified] => int
              [timecreated] => int
              [filesize] => int
              [author] => string
              [license] => string
              [filenameshort] => string
              [filesizeformatted] => string
              [icon] => string
              [timecreatedformatted] => string
              [timemodifiedformatted] => string
              [url] => string
            ]
          )
        )
      )
[tags] =>
  Array
    (
```





REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="discussions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timecreated">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="authorfullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="posts">
                        <SINGLE>
                            <KEY name="userposts">
                                <MULTIPLE>
                                    <SINGLE>
                                        <KEY name="id">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="subject">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="repliesubject">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="message">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="messageformat">
                                            <VALUE>int</VALUE>
                                        </KEY>
                                        <KEY name="author">
                                            <SINGLE>
                                                <KEY name="id">
                                                    <VALUE>int</VALUE>
                                                </KEY>
                                                <KEY name="fullname">
                                                    <VALUE>string</VALUE>
                                                </KEY>
                                                <KEY name="isdeleted">
                                                    <VALUE>int</VALUE>
                                                </KEY>
                                                <KEY name="groups">
                                                    <MULTIPLE>
                                                        <SINGLE>
                                                            <KEY name="id">
                                                                <VALUE>int</VALUE>
                                                            </KEY>
                                                            <KEY name="name">
                                                                <VALUE>string</VALUE>
                                                            </KEY>
                                                            <KEY name="urls">
                                                                <SINGLE>
                                                                    <KEY name="image">
                                                                        <VALUE>string</VALUE>
                                                                    </KEY>
                                                                </SINGLE>
                                                            </KEY>
                                                        </SINGLE>
                                                    </MULTIPLE>
                                                </KEY>
                                                <KEY name="urls">
                                                    <SINGLE>
                                                        <KEY name="profile">
                                                            <VALUE>string</VALUE>
                                                        </KEY>
                                                        <KEY name="profileimage">
                                                            <VALUE>string</VALUE>
                                                        </KEY>
                                                    </SINGLE>
                                                </KEY>
                                            </SINGLE>
                                        </KEY>
                                    </SINGLE>
                                </MULTIPLE>
                            </KEY>
                        </SINGLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="discussionid">
    </KEY>
</RESPONSE>

```

```
<VALUE>int</VALUE>
</KEY>
<KEY name="hasparent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="unread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isdeleted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isprivatereply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="haswordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="wordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="charcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="capabilities">
    <SINGLE>
        <KEY name="view">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="selfenrol">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="controlreadstatus">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplyprivately">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewisolated">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewparent">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
```



```
<KEY name="reply">
    <VALUE>string</VALUE>
</KEY>
<KEY name="export">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasunread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="discuss">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="attachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodifiedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="urls">
                <SINGLE>
                    <KEY name="export">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </KEY>
        </MULTIPLE>
    </KEY>
</KEY>
```



```
</KEY>
<KEY name="html">
    <SINGLE>
        <KEY name="plagiarism">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</MULTIPLE>
</KEY>
<KEY name="messageinlinefiles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodifiedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="tags">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="tagid">
                <VALUE>int</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

<KEY name="isstandard">
    <VALUE>int</VALUE>
</KEY>
<KEY name="displayname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="flag">
    <VALUE>int</VALUE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<MULTIPLE>
</KEY>
<KEY name="html">
    <SINGLE>
        <KEY name="rating">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="taglist">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="authorsubheading">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<SINGLE>
</KEY>
</MULTIPLE>
</KEY>
<KEY name="parentposts">
    <MULTIPLE>
        <SINGLE>
            <KEY name="id">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="subject">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="repliesubject">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="message">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="messageformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isdeleted">
                        <VALUE>int</VALUE>
                    </KEY>
                <SINGLE>
                    <KEY name="groups">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="id">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="name">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="urls">
                                    <SINGLE>
                                        <KEY name="image">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                    </SINGLE>
                                </KEY>
                            <SINGLE>
                                <KEY name="MULTIPLE">
                                    </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>

```



```
<KEY name="urls">
    <SINGLE>
        <KEY name="profile">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="profileimage">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="discussionid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="hasparent">
    <VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="unread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isdeleted">
    <VALUE>int</VALUE>
</KEY>
<KEY name="isprivatereply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="haswordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="wordcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="charcount">
    <VALUE>int</VALUE>
</KEY>
<KEY name="capabilities">
    <SINGLE>
        <KEY name="view">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="edit">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="delete">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="split">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="reply">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="selfenrol">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="export">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="controlreadstatus">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplyprivately">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewisolated">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
```



```
</KEY>
<KEY name="viewparent">
    <VALUE>string</VALUE>
</KEY>
<KEY name="edit">
    <VALUE>string</VALUE>
</KEY>
<KEY name="delete">
    <VALUE>string</VALUE>
</KEY>
<KEY name="split">
    <VALUE>string</VALUE>
</KEY>
<KEY name="reply">
    <VALUE>string</VALUE>
</KEY>
<KEY name="export">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="markasunread">
    <VALUE>string</VALUE>
</KEY>
<KEY name="discuss">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="attachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
```



```
</KEY>
<KEY name="timemodifiedformatted">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <VALUE>string</VALUE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="export">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="html">
    <SINGLE>
        <KEY name="plagiarism">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messageinlinefiles">
    <MULTIPLE>
        <SINGLE>
            <KEY name="contextid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="component">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filearea">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="itemid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isdir">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="isimage">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="timecreated">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="author">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="license">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filenameshort">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesizeformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="icon">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timecreatedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodifiedformatted">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <VALUE>string</VALUE>
```



```
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tags">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="tagid">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstandard">
<VALUE>int</VALUE>
</KEY>
<KEY name="displayname">
<VALUE>string</VALUE>
</KEY>
<KEY name="flag">
<VALUE>int</VALUE>
</KEY>
<KEY name="urls">
<SINGLE>
<KEY name="view">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="html">
<SINGLE>
<KEY name="rating">
<VALUE>string</VALUE>
</KEY>
<KEY name="taglist">
<VALUE>string</VALUE>
</KEY>
<KEY name="authorsubheading">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
<SINGLE>
<KEY name="item">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
<VALUE>string</VALUE>
</KEY>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`mod_forum_get_forum_access_information`

Return capabilities information for a given forum.

Arguments

`forumid` (Required)

Forum instance id.

General structure

```
int //Forum instance id.
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

Response

General structure

```

object {
  warnings Optional //List of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)canaddinstance int Optional //Whether the user has the capability mod/forum:addinstance allowed.
canviewdiscussion int Optional //Whether the user has the capability mod/forum:viewdiscussion allowed.
canviewhiddentimedposts int Optional //Whether the user has the capability mod/forum:viewhiddentimedposts allowed.
canstartdiscussion int Optional //Whether the user has the capability mod/forum:startdiscussion allowed.
canreplypost int Optional //Whether the user has the capability mod/forum:replypost allowed.
canaddnews int Optional //Whether the user has the capability mod/forum:addnews allowed.
canreplynews int Optional //Whether the user has the capability mod/forum:replynews allowed.
canviewrating int Optional //Whether the user has the capability mod/forum:viewrating allowed.
canviewanyrating int Optional //Whether the user has the capability mod/forum:viewanyrating allowed.
canviewallratings int Optional //Whether the user has the capability mod/forum:viewallratings allowed.
canrate int Optional //Whether the user has the capability mod/forum:rate allowed.
canpostprivatereply int Optional //Whether the user has the capability mod/forum:postprivatereply allowed.
canreadprivatereplies int Optional //Whether the user has the capability mod/forum:readprivatereplies allowed.
cancreateattachment int Optional //Whether the user has the capability mod/forum:createattachment allowed.
candeleteownpost int Optional //Whether the user has the capability mod/forum:deleteownpost allowed.
candeleteanypost int Optional //Whether the user has the capability mod/forum:deleteanypost allowed.
cansplitdiscussions int Optional //Whether the user has the capability mod/forum:splittdiscussions allowed.
canmovediscussions int Optional //Whether the user has the capability mod/forum:movediscussions allowed.
canpindiscussions int Optional //Whether the user has the capability mod/forum:pindiscussions allowed.
caneditanypost int Optional //Whether the user has the capability mod/forum:editanypost allowed.
canviewqandawithoutposting int Optional //Whether the user has the capability mod/forum:viewqandawithoutposting allowed.
canviewsubscribers int Optional //Whether the user has the capability mod/forum:viewsubscribers allowed.
canmanagesubscriptions int Optional //Whether the user has the capability mod/forum:managesubscriptions allowed.
canpostwithouthrottling int Optional //Whether the user has the capability mod/forum:postwithouthrottling allowed.
canexportdiscussion int Optional //Whether the user has the capability mod/forum:exportdiscussion allowed.
canexportforum int Optional //Whether the user has the capability mod/forum:exportforum allowed.
canexportpost int Optional //Whether the user has the capability mod/forum:exportpost allowed.
canexportownpost int Optional //Whether the user has the capability mod/forum:exportownpost allowed.
canaddquestion int Optional //Whether the user has the capability mod/forum:addquestion allowed.
canallowforcesubscribe int Optional //Whether the user has the capability mod/forum:allowforcesubscribe allowed.
cancanposttomygroups int Optional //Whether the user has the capability mod/forum:canposttomygroups allowed.
cancanoverridediscussionlock int Optional //Whether the user has the capability mod/forum:canoverridediscussionlock allowed.
cancanoverridecutoff int Optional //Whether the user has the capability mod/forum:canoverridecutoff allowed.
cancantogglefavourite int Optional //Whether the user has the capability mod/forum:cantogglefavourite allowed.
cangrade int Optional //Whether the user has the capability mod/forum:grade allowed.
}

```



XML-RPC (PHP structure)

```
Array
(
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
[canaddinstance] => int
[canviewdiscussion] => int
[canviewhiddentimedposts] => int
[canstartdiscussion] => int
[canreplypost] => int
[canaddnews] => int
[canreplynews] => int
[canviewrating] => int
[canviewanyrating] => int
[canviewallratings] => int
[canrate] => int
[canpostprivatereply] => int
[canreadprivatereplies] => int
[cancreateattachment] => int
[candeleteownpost] => int
[candeleteanypost] => int
[cansplitdiscussions] => int
[camovediscussions] => int
[canpindiscussions] => int
[caneditanypost] => int
[canviewqandawithoutposting] => int
[canviewsubscribers] => int
[canmanagesubscriptions] => int
[canpostwithoutthrottling] => int
[canexportdiscussion] => int
[canexportforum] => int
[canexportpost] => int
[canexportownpost] => int
[canaddquestion] => int
[canallowforcesubscribe] => int
[cancancelposttomygroups] => int
[canoverridediscussionlock] => int
[canoverridecutoff] => int
[canctogglefavourite] => int
[cangrade] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="canaddinstance">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewdiscussion">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewhiddentimedposts">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canstartdiscussion">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplypost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canaddnews">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreplynews">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewrating">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewanyrating">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canviewallratings">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canrate">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpostprivateresponse">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canreadprivateresponses">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cancreateattachment">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="candeleteownpost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="candeleteanypost">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="cansplitdiscussions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="camovediscussions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="canpin discussions">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="caneditanypost">
            <VALUE>int</VALUE>
        </KEY>
```



```

<KEY name="canviewqandawithoutposting">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canviewsubscribers">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canmanagesubscriptions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canpostwithoutthrottling">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canexportdiscussion">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canexportforum">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canexportpost">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canexporttownpost">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canaddquestion">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canallowforcesubscribe">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cancanposttomygroups">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cancanoverridediscussionlock">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cancanoverridecutoff">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cantogglefavourite">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cangrade">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_get_forum_discussions

Returns a list of forum discussions optionally sorted and paginated.

Arguments

forumid (Required)

forum instance id

General structure

```
int //forum instance id
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

sortorder (Default to "-1")

sort by this element: numreplies,, created or timemodified

General structure

```
int Default to "-1" //sort by this element: numreplies, , created or timemodified
```

XML-RPC (PHP structure)

```
[sortorder] => int
```

REST (POST parameters)

```
sortorder= int
```

page (Default to "-1")

current page

General structure

```
int Default to "-1" //current page
```

XML-RPC (PHP structure)

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

perpage (Default to "0")

items per page

General structure

```
int Default to "0" //items per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)`perpage= int``groupid (Default to "0")``group id`**General structure**`int Default to "0" //group id`**XML-RPC (PHP structure)**`[groupid] => int`**REST (POST parameters)**`groupid= int`**Response**

```

General structure

object {
  discussions list of (
    //post
  object {
    id int //Post id
    name string //Discussion name
    groupid int //Group id
    timemodified int //Time modified
    usermodified int //The id of the user who last modified
    timestampstart int //Time discussion can start
    timeend int //Time discussion ends
    discussion int //Discussion id
    parent int //Parent id
    userid int //User who started the discussion id
    created int //Creation time
    modified int //Time modified
    mailed int //Mailed?
    subject string //The post subject
    message string //The post message
    messageformat int //Message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
    messagetrust int //Can we trust?
    messageinlinefiles optional //post message inline files
    list of (
      //File
    object {
      filename string optional //File name.
      filepath string optional //File path.
      filesize int optional //File size.
      fileurl string optional //Downloadable file url.
      timemodified int optional //Time modified.
      mimetype string optional //File mime type.
      isexternalfile int optional //Whether is an external file.
      repositorytype string optional //The repository type for external files.
    }
  )attachment string //Has attachments?
  attachments optional //attachments
  list of (
    //File
  object {
    filename string optional //File name.
    filepath string optional //File path.
    filesize int optional //File size.
    fileurl string optional //Downloadable file url.
    timemodified int optional //Time modified.
    mimetype string optional //File mime type.
    isexternalfile int optional //Whether is an external file.
    repositorytype string optional //The repository type for external files.
  }
)totalscore int //The post message total score
mailnow int //Mail now?
userfullname string //Post author full name
usermodifiedfullname string //Post modifier full name
userpictureurl string //Post author picture.
usermodifiedpictureurl string //Post modifier picture.
numreplies int //The number of replies in the discussion
numunread int //The number of unread discussions.
pinned int //Is the discussion pinned
locked int //Is the discussion locked
starred int //Is the discussion starred
canreply int //Can the user reply to the discussion
canlock int //Can the user lock the discussion
canfavourite int //Can the user star the discussion
}
)warnings optional //List of warnings
list of (
  //warning
  object {
    item string optional //item
    itemid int optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
}

```



XML-RPC (PHP structure)

```

Array
(
    [discussions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [groupid] => int
                            [timemodified] => int
                            [usermodified] => int
                            [timestart] => int
                            [timeend] => int
                            [discussion] => int
                            [parent] => int
                            [userid] => int
                            [created] => int
                            [modified] => int
                            [mailed] => int
                            [subject] => string
                            [message] => string
                            [messageformat] => int
                            [messagetrust] => int
                            [messageinlinefiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [attachment] => string
                            [attachments] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [totalscore] => int
                            [mailnow] => int
                            [userfullname] => string
                            [usermodifiedfullname] => string
                            [userpictureurl] => string
                            [usermodifiedpictureurl] => string
                            [numreplies] => int
                            [numunread] => int
                            [pinned] => int
                            [locked] => int
                            [starred] => int
                            [canreply] => int
                            [canlock] => int
                            [canfavourite] => int
                        )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (

```



```
    [item] => string
    [itemid] => int
    [warningcode] => string
    [message] => string
)
)
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="discussions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usermodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeend">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="discussion">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="parent">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="created">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="modified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mailed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="messageformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="messagetrust">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="messageinlinefiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</MULTIPLE>
</KEY>
<KEY name="attachment">
    <VALUE>string</VALUE>
</KEY>
<KEY name="attachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="totalscore">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mailnow">
    <VALUE>int</VALUE>
</KEY>
<KEY name="userfullname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usermodifiedfullname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="userpictureurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usermodifiedpictureurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="numreplies">
    <VALUE>int</VALUE>
</KEY>
<KEY name="numunread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="pinned">
    <VALUE>int</VALUE>
</KEY>
<KEY name="locked">
    <VALUE>int</VALUE>
</KEY>
<KEY name="starred">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canreply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canlock">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canfavourite">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
```



```

</KEY>
<KEY name="warnings">
  <MULTIPLE>
    <SINGLE>
      <KEY name="item">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="itemid">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="warningcode">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="message">
        <VALUE>string</VALUE>
      </KEY>
    </SINGLE>
  </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
  <MESSAGE>Invalid parameter value detected</MESSAGE>
  <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_get_forum_discussions_paginated Deprecated

**** DEPRECATED **** Please do not call this function any more. Returns a list of forum discussions optionally sorted and paginated.



Arguments

forumid (Required)

forum instance id

General structure

```
int //forum instance id
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

sortby (Default to "timemodified")

sort by this element: id, timemodified, timestamp or timeend

General structure

```
string Default to "timemodified" //sort by this element: id, timemodified, timestamp or timeend
```

XML-RPC (PHP structure)

```
[sortby] => string
```

REST (POST parameters)

```
sortby= string
```

sortdirection (Default to "DESC")

sort direction: ASC or DESC

General structure

```
string Default to "DESC" //sort direction: ASC or DESC
```

XML-RPC (PHP structure)

```
[sortdirection] => string
```

REST (POST parameters)

```
sortdirection= string
```

page (Default to "-1")

current page

General structure

```
int Default to "-1" //current page
```

XML-RPC (PHP structure)

```
[page] => int
```

REST (POST parameters)

```
page= int
```



perpage (Default to "0")

items per page

General structure

```
int Default to "0" //items per page
```

XML-RPC (PHP structure)

```
[perpage] => int
```

REST (POST parameters)

```
perpage= int
```

Response

```

General structure

object {
  discussions list of (
    //post
  object {
    id int //Post id
    name string //Discussion name
    groupid int //Group id
    timemodified int //Time modified
    usermodified int //The id of the user who Last modified
    timestamp int //Time discussion can start
    timeend int //Time discussion ends
    discussion int //Discussion id
    parent int //Parent id
    userid int //User who started the discussion id
    created int //Creation time
    modified int //Time modified
    mailed int //Mailed?
    subject string //The post subject
    message string //The post message
    messageformat int //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
    messagetrust int //Can we trust?
    messageinlinefiles optional //post message inline files
    list of (
      //File.
    object {
      filename string Optional //File name.
      filepath string Optional //File path.
      filesize int Optional //File size.
      fileurl string Optional //Downloadable file url.
      timemodified int Optional //Time modified.
      mimetype string Optional //File mime type.
      isexternalfile int Optional //Whether is an external file.
      repositorytype string Optional //The repository type for external files.
    }
  )attachment string //Has attachments?
  attachments optional //attachments
  list of (
    //File.
  object {
    filename string Optional //File name.
    filepath string Optional //File path.
    filesize int Optional //File size.
    fileurl string Optional //Downloadable file url.
    timemodified int Optional //Time modified.
    mimetype string Optional //File mime type.
    isexternalfile int Optional //Whether is an external file.
    repositorytype string Optional //The repository type for external files.
  }
)totalscore int //The post message total score
mailnow int //Mail now?
userfullname string //Post author full name
usermodifiedfullname string //Post modifier full name
userpictureurl string //Post author picture.
usermodifiedpictureurl string //Post modifier picture.
numreplies int //The number of replies in the discussion
numunread int //The number of unread discussions.
pinned int //Is the discussion pinned
locked int //Is the discussion Locked
canreply int //Can the user reply to the discussion
canlock int //Can the user Lock the discussion
)
)warnings optional //list of warnings
list of (
  //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
}

```



XML-RPC (PHP structure)

```

Array
(
    [discussions] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [name] => string
                            [groupid] => int
                            [timemodified] => int
                            [usermodified] => int
                            [timestart] => int
                            [timeend] => int
                            [discussion] => int
                            [parent] => int
                            [userid] => int
                            [created] => int
                            [modified] => int
                            [mailed] => int
                            [subject] => string
                            [message] => string
                            [messageformat] => int
                            [messagetrust] => int
                            [messageinlinefiles] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [attachment] => string
                            [attachments] =>
                                Array
                                    (
                                        [0] =>
                                            Array
                                                (
                                                    [filename] => string
                                                    [filepath] => string
                                                    [filesize] => int
                                                    [fileurl] => string
                                                    [timemodified] => int
                                                    [mimetype] => string
                                                    [isexternalfile] => int
                                                    [repositorytype] => string
                                                )
                                    )
                            )
                            [totalscore] => int
                            [mailnow] => int
                            [userfullname] => string
                            [usermodifiedfullname] => string
                            [userpictureurl] => string
                            [usermodifiedpictureurl] => string
                            [numreplies] => int
                            [numunread] => int
                            [pinned] => int
                            [locked] => int
                            [canreply] => int
                            [canlock] => int
                        )
                )
            )
        )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                        )
            )
)

```



```
[warningcode] => string  
[message] => string  
)  
)
```

^

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="discussions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="groupid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="usermodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timestart">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="timeend">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="discussion">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="parent">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="created">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="modified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mailed">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="subject">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="messageformat">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="messagetrust">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="messageinlinefiles">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filepath">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="filesize">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="isexternalfile">
                                    <VALUE>int</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



```
<KEY name="repositorytype">
    <VALUE>string</VALUE>
</KEY>
</MULTIPLE>
</KEY>
<KEY name="attachment">
    <VALUE>string</VALUE>
</KEY>
<KEY name="attachments">
    <MULTIPLE>
        <SINGLE>
            <KEY name="filename">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
<KEY name="totalscore">
    <VALUE>int</VALUE>
</KEY>
<KEY name="mailnow">
    <VALUE>int</VALUE>
</KEY>
<KEY name="userfullname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usermodifiedfullname">
    <VALUE>string</VALUE>
</KEY>
<KEY name="userpictureurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="usermodifiedpictureurl">
    <VALUE>string</VALUE>
</KEY>
<KEY name="numreplies">
    <VALUE>int</VALUE>
</KEY>
<KEY name="numunread">
    <VALUE>int</VALUE>
</KEY>
<KEY name="pinned">
    <VALUE>int</VALUE>
</KEY>
<KEY name="locked">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canreply">
    <VALUE>int</VALUE>
</KEY>
<KEY name="canlock">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
    <MULTIPLE>
        <SINGLE>
            <KEY name="item">
                <VALUE>string</VALUE>
            </KEY>
        </SINGLE>
    </MULTIPLE>
</KEY>
```



```

</KEY>
<KEY name="itemid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
    <VALUE>string</VALUE>
</KEY>
<KEY name="message">
    <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_get_forums_by_courses

Returns a list of forum instances in a provided set of courses, if no courses are provided then all the forum instances the user has access to will be returned.

Arguments

courseids (Default to "Array ()")

Array of Course IDs



General structure

Default to "Array
(
)
" //Array of Course IDs
list of (
int //course ID
)

XML-RPC (PHP structure)

[courseids] =>
Array
(
[0] => int
)

REST (POST parameters)

courseids[0]= int

Response

```
General structure

list of (
    //forum
object {
id int //Forum id
course int //Course id
type string //The forum type
name string //Forum name
intro string //The forum intro
introformat int //intro format (1 = HTML, 0 = MOODLE, or 4 = MARKDOWN
introfiles Optional //Files in the introduction text
list of (
    //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)lang string Optional //Forced activity language
duedate int Optional //duedate for the user
cutoffdate int Optional //cutoffdate for the user
assessed int //Aggregate type
assesstimestart int //Assess start time
assesstimefinish int //Assess finish time
scale int //Scale
grade_forum int //Whole forum grade
grade_forum_notify int //Whether to send notifications to students upon grading by default
maxbytes int //Maximum attachment size
maxattachments int //Maximum number of attachments
forcesubscribe int //Force users to subscribe
trackingtype int //Subscription mode
rsstype int //RSS feed for this activity
rssarticles int //Number of RSS recent articles
timemodified int //Time modified
warnafter int //Post threshold for warning
blockafter int //Post threshold for blocking
blockperiod int //Time period for blocking
completiondiscussions int //Student must create discussions
completionreplies int //Student must post replies
completionposts int //Student must post discussions or replies
cmid int //Course module id
numdiscussions int Optional //Number of discussions in the forum
cancreatediscussions int Optional //If the user can create discussions
lockdiscussionafter int Optional //After what period a discussion is locked
istracked int Optional //If the user is tracking the forum
unreadpostscount int Optional //The number of unread posts for tracked forums
}
)
```



XML-RPC (PHP structure)

```
Array
(
    [
        [0] =>
            Array
            (
                [
                    [id] => int
                    [course] => int
                    [type] => string
                    [name] => string
                    [intro] => string
                    [introformat] => int
                    [introfiles] =>
                        Array
                        (
                            [
                                [0] =>
                                    Array
                                    (
                                        [
                                            [filename] => string
                                            [filepath] => string
                                            [filesize] => int
                                            [fileurl] => string
                                            [timemodified] => int
                                            [mimetype] => string
                                            [isexternalfile] => int
                                            [repositorytype] => string
                                        )
                                    )
                            )
                ]
                [
                    [lang] => string
                    [duedate] => int
                    [cutoffdate] => int
                    [assessed] => int
                    [assesstimestart] => int
                    [assesstimefinish] => int
                    [scale] => int
                    [grade_forum] => int
                    [grade_forum_notify] => int
                    [maxbytes] => int
                    [maxattachments] => int
                    [forcesubscribe] => int
                    [trackingtype] => int
                    [rsstype] => int
                    [rssarticles] => int
                    [timemodified] => int
                    [warnafter] => int
                    [blockafter] => int
                    [blockperiod] => int
                    [completiondiscussions] => int
                    [completionreplies] => int
                    [completionposts] => int
                    [cmid] => int
                    [numdiscussions] => int
                    [cancreatediscussions] => int
                    [lockdiscussionafter] => int
                    [istracked] => int
                    [unreadpostscount] => int
                )
            )
    )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <MULTIPLE>
    <SINGLE>
      <KEY name="id">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="course">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="type">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="name">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="intro">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="introformat">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="introfiles">
        <MULTIPLE>
          <SINGLE>
            <KEY name="filename">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="filepath">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="filesize">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="fileurl">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="timemodified">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="mimetype">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="isexternalfile">
              <VALUE>int</VALUE>
            </KEY>
            <KEY name="repositorytype">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </MULTIPLE>
      </KEY>
      <KEY name="lang">
        <VALUE>string</VALUE>
      </KEY>
      <KEY name="duedate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="cutoffdate">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="assessed">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="assesstimestart">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="assesstimefinish">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="scale">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="grade_forum">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="grade_forum_notify">
        <VALUE>int</VALUE>
      </KEY>
      <KEY name="maxbytes">
        <VALUE>int</VALUE>
      </KEY>
```



```

</KEY>
<KEY name="maxattachments">
    <VALUE>int</VALUE>
</KEY>
<KEY name="forcesubscribe">
    <VALUE>int</VALUE>
</KEY>
<KEY name="trackingtype">
    <VALUE>int</VALUE>
</KEY>
<KEY name="rsstype">
    <VALUE>int</VALUE>
</KEY>
<KEY name="rssarticles">
    <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
    <VALUE>int</VALUE>
</KEY>
<KEY name="warnafter">
    <VALUE>int</VALUE>
</KEY>
<KEY name="blockafter">
    <VALUE>int</VALUE>
</KEY>
<KEY name="blockperiod">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completiondiscussions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completionreplies">
    <VALUE>int</VALUE>
</KEY>
<KEY name="completionposts">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cmid">
    <VALUE>int</VALUE>
</KEY>
<KEY name="numdiscussions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="cancreatediscussions">
    <VALUE>int</VALUE>
</KEY>
<KEY name="lockdiscussionafter">
    <VALUE>int</VALUE>
</KEY>
<KEY name="istracked">
    <VALUE>int</VALUE>
</KEY>
<KEY name="unreadpostscount">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>

```



Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_prepare_draft_area_for_post

Prepares a draft area for editing a post.

Arguments**postid** (Required)

Post to prepare the draft area for.

General structure

```
int //Post to prepare the draft area for.
```

XML-RPC (PHP structure)

```
[postid] => int
```

REST (POST parameters)

```
postid= int
```

area (Required)

Area to prepare: attachment or post.

General structure

```
string //Area to prepare: attachment or post.
```

XML-RPC (PHP structure)

```
[area] => string
```

REST (POST parameters)

```
area= string
```

draftitemid (Default to "0")

The draft item id to use. 0 to generate one.

**General structure**

```
int Default to "0" //The draft item id to use. 0 to generate one.
```

XML-RPC (PHP structure)

```
[draftitemid] => int
```

REST (POST parameters)

```
draftitemid= int
```

filestokeep (Default to "Array ()")

Only keep these files in the draft file area. Empty for keeping all.

```
General structure

Default to "Array
(
)
" //Only keep these files in the draft file area. Empty for keeping all.
list of (
object {
filename string  //File name.
filepath string  //File path.
}
)
```

XML-RPC (PHP structure)

```
[filestokeep] =>
Array
(
[0] =>
Array
(
[filestokeep] => string
[filepath] => string
)
)
```

REST (POST parameters)

```
filestokeep[0][filename]= string
filestokeep[0][filepath]= string
```

Response

```
General structure

object {
draftitemid int  //Draft item id for the file area.
files  optional //Draft area files.
list of (
//File.
object {
filename string  optional //File name.
filepath string  optional //File path.
filesize int  optional //File size.
fileurl string  optional //Downloadable file url.
timemodified int  optional //Time modified.
mimetype string  optional //File mime type.
isexternalfile int  optional //Whether is an external file.
repositorytype string  optional //The repository type for external files.
}
)areaoptions  //Draft file area options.
list of (
object {
name string  //Name of option.
value string  //Value of option.
}
)messagetext string  //Message text with URLs rewritten.
warnings  optional //List of warnings
list of (
//warning
object {
item string  optional //item
itemid int  optional //item id
warningcode string  //the warning code can be used by the client app to implement specific behaviour
message string  //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [draftitemid] => int
    [files] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [filename] => string
                            [filepath] => string
                            [filesize] => int
                            [fileurl] => string
                            [timemodified] => int
                            [mimetype] => string
                            [isexternalfile] => int
                            [repositorytype] => string
                        )
                )
)
[areaoptions] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
[messagetext] => string
[warnings] =>
    Array
        (
            [0] =>
                Array
                    (
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
            )
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="draftitemid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="files">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="filename">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filepath">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="filesize">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fileurl">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="timemodified">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mimetype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="isexternalfile">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="repositorytype">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="areaoptions">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="value">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="messagetext">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```



Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_set_lock_state

Set the lock state for the discussion

Arguments**forumid** (Required)

Forum that the discussion is in

General structure

```
int //Forum that the discussion is in
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

discussionid (Required)

The discussion to lock / unlock

General structure

```
int //The discussion to Lock / unlock
```

**XML-RPC (PHP structure)**

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

targetstate (Required)

The timestamp for the lock state

General structure

```
int //The timestamp for the Lock state
```

XML-RPC (PHP structure)

```
[targetstate] => int
```

REST (POST parameters)

```
targetstate= int
```

Response**General structure**

```
object {
    id int //The discussion we are Locking.
    locked int //The Locked state of the discussion.
    times object {
        locked int //The Locked time of the discussion.
    }
}
```

XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [locked] => int
    [times] =>
        Array
        (
            [locked] => int
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="id">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="locked">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="times">
            <SINGLE>
                <KEY name="locked">
                    <VALUE>int</VALUE>
                </KEY>
            </SINGLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`mod_forum_set_pin_state`

Set the pin state

Arguments

`discussionid` (Required)

The discussion to pin or unpin

General structure

```
int //The discussion to pin or unpin
```

XML-RPC (PHP structure)

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

`targetstate` (Required)

The target state

General structure

```
int //The target state
```

XML-RPC (PHP structure)

```
[targetstate] => int
```

REST (POST parameters)

```
targetstate= int
```



Response

General structure

```
object {
id int //id
forumid int //forumid
pinned int //pinned
locked int //Locked
istimedlocked int //istimedlocked
name string //name
firstpostid int //firstpostid
group object {
name string //name
urls object {
picture string Optional //picture
userlist string Optional //userlist
}
}
times object {
modified int //modified
start int //start
end int //end
locked int //Locked
}
userstate object {
subscribed int //subscribed
favourited int //favourited
}
capabilities object {
subscribe int //subscribe
move int //move
pin int //pin
post int //post
manage int //manage
favourite int //favourite
}
urls object {
view string //view
viewlatest string Optional //viewlatest
viewfirstunread string Optional //viewfirstunread
markasread string //markasread
subscribe string //subscribe
pin string Optional //pin
}
timed object {
istimed int Optional //istimed
visible int Optional //visible
}
}
```



XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [forumid] => int
    [pinned] => int
    [locked] => int
    [istimedlocked] => int
    [name] => string
    [firstpostid] => int
    [group] =>
        Array
            (
                [name] => string
                [urls] =>
                    Array
                        (
                            [picture] => string
                            [userlist] => string
                        )
                )
            )
    [times] =>
        Array
            (
                [modified] => int
                [start] => int
                [end] => int
                [locked] => int
            )
    [userstate] =>
        Array
            (
                [subscribed] => int
                [favourited] => int
            )
    [capabilities] =>
        Array
            (
                [subscribe] => int
                [move] => int
                [pin] => int
                [post] => int
                [manage] => int
                [favourite] => int
            )
    [urls] =>
        Array
            (
                [view] => string
                [viewlatest] => string
                [viewfirstunread] => string
                [markasread] => string
                [subscribe] => string
                [pin] => string
            )
    [timed] =>
        Array
            (
                [istimed] => int
                [visible] => int
            )
)
^
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="forumid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pinned">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="istimelocked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="firstpostid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="group">
      <SINGLE>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="urls">
          <SINGLE>
            <KEY name="picture">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="userlist">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="times">
  <SINGLE>
    <KEY name="modified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="start">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="end">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="userstate">
  <SINGLE>
    <KEY name="subscribed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="favourited">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="capabilities">
  <SINGLE>
    <KEY name="subscribe">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="move">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pin">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="post">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
```



```

<KEY name="manage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="favourite">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewlatest">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewfirstunread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscribe">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="pin">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="timed">
    <SINGLE>
        <KEY name="istimed">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_forum_set_subscription_state

Set the subscription state

Arguments

forumid (Required)

Forum that the discussion is in

General structure

```
int //Forum that the discussion is in
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

discussionid (Required)

The discussion to subscribe or unsubscribe

General structure

```
int //The discussion to subscribe or unsubscribe
```

XML-RPC (PHP structure)

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

targetstate (Required)

The target state

General structure

```
int //The target state
```

XML-RPC (PHP structure)

```
[targetstate] => int
```

REST (POST parameters)

```
targetstate= int
```

**Response**

General structure

```
object {
id int //id
forumid int //forumid
pinned int //pinned
locked int //Locked
istimedlocked int //istimedlocked
name string //name
firstpostid int //firstpostid
group object {
name string //name
urls object {
picture string Optional //picture
userlist string Optional //userlist
}
}
times object {
modified int //modified
start int //start
end int //end
locked int //Locked
}
userstate object {
subscribed int //subscribed
favourited int //favourited
}
capabilities object {
subscribe int //subscribe
move int //move
pin int //pin
post int //post
manage int //manage
favourite int //favourite
}
urls object {
view string //view
viewlatest string Optional //viewlatest
viewfirstunread string Optional //viewfirstunread
markasread string //markasread
subscribe string //subscribe
pin string Optional //pin
}
timed object {
istimed int Optional //istimed
visible int Optional //visible
}
}
```



XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [forumid] => int
    [pinned] => int
    [locked] => int
    [istimedlocked] => int
    [name] => string
    [firstpostid] => int
    [group] =>
        Array
            (
                [name] => string
                [urls] =>
                    Array
                        (
                            [picture] => string
                            [userlist] => string
                        )
            )
    [times] =>
        Array
            (
                [modified] => int
                [start] => int
                [end] => int
                [locked] => int
            )
    [userstate] =>
        Array
            (
                [subscribed] => int
                [favourited] => int
            )
    [capabilities] =>
        Array
            (
                [subscribe] => int
                [move] => int
                [pin] => int
                [post] => int
                [manage] => int
                [favourite] => int
            )
    [urls] =>
        Array
            (
                [view] => string
                [viewlatest] => string
                [viewfirstunread] => string
                [markasread] => string
                [subscribe] => string
                [pin] => string
            )
)
[timed] =>
    Array
        (
            [istimed] => int
            [visible] => int
        )
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="forumid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pinned">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="istimelocked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="firstpostid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="group">
      <SINGLE>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="urls">
          <SINGLE>
            <KEY name="picture">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="userlist">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="times">
  <SINGLE>
    <KEY name="modified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="start">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="end">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="userstate">
  <SINGLE>
    <KEY name="subscribed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="favourited">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="capabilities">
  <SINGLE>
    <KEY name="subscribe">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="move">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pin">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="post">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
```



```

<KEY name="manage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="favourite">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewlatest">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewfirstunread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscribe">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="pin">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="timed">
    <SINGLE>
        <KEY name="istimed">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</KEY>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

`mod_forum_toggle_favourite_state`

Toggle the favourite state

Arguments

`discussionid` (Required)

The discussion to subscribe or unsubscribe

General structure

```
int //The discussion to subscribe or unsubscribe
```

XML-RPC (PHP structure)

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

targetstate (Required)

The target state

General structure

```
int //The target state
```

XML-RPC (PHP structure)

```
[targetstate] => int
```

REST (POST parameters)

```
targetstate= int
```

Response



General structure

```
object {
id int //id
forumid int //forumid
pinned int //pinned
locked int //Locked
istimedlocked int //istimedlocked
name string //name
firstpostid int //firstpostid
group object {
name string //name
urls object {
picture string Optional //picture
userlist string Optional //userlist
}
}
times object {
modified int //modified
start int //start
end int //end
locked int //Locked
}
userstate object {
subscribed int //subscribed
favourited int //favourited
}
capabilities object {
subscribe int //subscribe
move int //move
pin int //pin
post int //post
manage int //manage
favourite int //favourite
}
urls object {
view string //view
viewlatest string Optional //viewlatest
viewfirstunread string Optional //viewfirstunread
markasread string //markasread
subscribe string //subscribe
pin string Optional //pin
}
timed object {
istimed int Optional //istimed
visible int Optional //visible
}
}
```



XML-RPC (PHP structure)

```
Array
(
    [id] => int
    [forumid] => int
    [pinned] => int
    [locked] => int
    [istimedlocked] => int
    [name] => string
    [firstpostid] => int
    [group] =>
        Array
            (
                [name] => string
                [urls] =>
                    Array
                        (
                            [picture] => string
                            [userlist] => string
                        )
                )
            )
    [times] =>
        Array
            (
                [modified] => int
                [start] => int
                [end] => int
                [locked] => int
            )
    [userstate] =>
        Array
            (
                [subscribed] => int
                [favourited] => int
            )
    [capabilities] =>
        Array
            (
                [subscribe] => int
                [move] => int
                [pin] => int
                [post] => int
                [manage] => int
                [favourite] => int
            )
    [urls] =>
        Array
            (
                [view] => string
                [viewlatest] => string
                [viewfirstunread] => string
                [markasread] => string
                [subscribe] => string
                [pin] => string
            )
    [timed] =>
        Array
            (
                [istimed] => int
                [visible] => int
            )
)
^
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
  <SINGLE>
    <KEY name="id">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="forumid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pinned">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="istimelocked">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="name">
      <VALUE>string</VALUE>
    </KEY>
    <KEY name="firstpostid">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="group">
      <SINGLE>
        <KEY name="name">
          <VALUE>string</VALUE>
        </KEY>
        <KEY name="urls">
          <SINGLE>
            <KEY name="picture">
              <VALUE>string</VALUE>
            </KEY>
            <KEY name="userlist">
              <VALUE>string</VALUE>
            </KEY>
          </SINGLE>
        </KEY>
      </SINGLE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="times">
  <SINGLE>
    <KEY name="modified">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="start">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="end">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="locked">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="userstate">
  <SINGLE>
    <KEY name="subscribed">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="favourited">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
<KEY name="capabilities">
  <SINGLE>
    <KEY name="subscribe">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="move">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="pin">
      <VALUE>int</VALUE>
    </KEY>
    <KEY name="post">
      <VALUE>int</VALUE>
    </KEY>
  </SINGLE>
</KEY>
```



```

<KEY name="manage">
    <VALUE>int</VALUE>
</KEY>
<KEY name="favourite">
    <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="urls">
    <SINGLE>
        <KEY name="view">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewlatest">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="viewfirstunread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="markasread">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="subscribe">
            <VALUE>string</VALUE>
        </KEY>
        <KEY name="pin">
            <VALUE>string</VALUE>
        </KEY>
    </SINGLE>
</KEY>
<KEY name="timed">
    <SINGLE>
        <KEY name="istimed">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="visible">
            <VALUE>int</VALUE>
        </KEY>
    </SINGLE>
</KEY>
</SINGLE>
</RESPONSE>

```

Error message

REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod_forum_update_discussion_post

Updates a post or a discussion topic post.

Arguments

postid (Required)

Post to be updated. It can be a discussion topic post.

General structure

```
int //Post to be updated. It can be a discussion topic post.
```

XML-RPC (PHP structure)

```
[postid] => int
```

REST (POST parameters)

```
postid= int
```

subject (Default to "")

Updated post subject

General structure

```
string Default to "" //Updated post subject
```

XML-RPC (PHP structure)

```
[subject] => string
```

REST (POST parameters)

```
subject= string
```

message (Default to "")

Updated post message (HTML assumed if messageformat is not provided)

General structure

```
string Default to "" //Updated post message (HTML assumed if messageformat is not provided)
```

XML-RPC (PHP structure)

```
[message] => string
```

REST (POST parameters)

```
message= string
```



messageformat (Default to "1")

message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)

General structure

```
int Default to "1" //message format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
```

XML-RPC (PHP structure)

```
[messageformat] => int
```

REST (POST parameters)

```
messageformat= int
```

options (Default to "Array ()")

Configuration options for the post.

```
General structure

Default to "Array
(
)
" //configuration options for the post.
list of (
object {
name string //The allowed keys (value format) are:
            pinned (bool); (only for discussions) whether to pin this discussion or not
            discussionsubscribe (bool); whether to subscribe to the post or not
            inlineattachmentsid (int); the draft file area id for inline attachments in the text
            attachmentsid (int); the draft file area id for attachments
value string //The value of the option.
}
)
```

XML-RPC (PHP structure)

```
[options] =>
    Array
        (
            [0] =>
                Array
                    (
                        [name] => string
                        [value] => string
                    )
        )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response

```
General structure

object {
status int //True if the post/discussion was updated, false otherwise.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)}
```

**XML-RPC (PHP structure)**

```
Array
(
    [status] => int
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_forum_view_forum**

Trigger the course module viewed event and update the module completion status.

Arguments**forumid** (Required)

forum instance id

General structure

```
int //forum instance id
```

XML-RPC (PHP structure)

```
[forumid] => int
```

REST (POST parameters)

```
forumid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
    ) object {
        item string Optional //item
        itemid int Optional //item id
        warningcode string //the warning code can be used by the client app to implement specific behaviour
        message string //untranslated english message to explain the warning
    }
})}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

**Error message**

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_forum_view_forum_discussion

Trigger the forum discussion viewed event.

Arguments**discussionid** (Required)

discussion id

General structure

```
int //discussion id
```

XML-RPC (PHP structure)

```
[discussionid] => int
```

REST (POST parameters)

```
discussionid= int
```

Response**General structure**

```
object {
    status int //status: true if success
    warnings Optional //list of warnings
    list of (
        //warning
        object {
            item string Optional //item
            itemid int Optional //item id
            warningcode string //the warning code can be used by the client app to implement specific behaviour
            message string //untranslated english message to explain the warning
        }
    )}
```

XML-RPC (PHP structure)

```
Array
(
    [status] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="status">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_glossary_add_entry

Add a new entry to a given glossary

Arguments

glossaryid (Required)

Glossary id

General structure

```
int //Glossary id
```

XML-RPC (PHP structure)

```
[glossaryid] => int
```

REST (POST parameters)

```
glossaryid= int
```

concept (Required)

Glossary concept

General structure

```
string //Glossary concept
```

XML-RPC (PHP structure)

```
[concept] => string
```

REST (POST parameters)

```
concept= string
```

definition (Required)

Glossary concept definition

General structure

```
string //Glossary concept definition
```



XML-RPC (PHP structure)

```
[definition] => string
```

REST (POST parameters)

```
definition= string
```

definitionformat (Required)

definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)

General structure

```
int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
```

XML-RPC (PHP structure)

```
[definitionformat] => int
```

REST (POST parameters)

```
definitionformat= int
```

options (Default to "Array ()")

Optional settings

General structure

```
Default to "Array"
(
)
" //Optional settings
list of (
object {
name string //The allowed keys (value format) are:
    inlineattachmentsid (int); the draft file area id for inline attachments
    attachmentsid (int); the draft file area id for attachments
    categories (comma separated int); comma separated category ids
    aliases (comma separated str); comma separated aliases
    usedynalink (bool); whether the entry should be automatically Linked.
    casesensitive (bool); whether the entry is case sensitive.
    fullmatch (bool); whether to match whole words only.
value string //the value of the option (validated inside the function)
}
)
```

XML-RPC (PHP structure)

```
[options] =>
Array
(
[0] =>
    Array
    (
        [name] => string
        [value] => string
    )
)
```

REST (POST parameters)

```
options[0][name]= string
options[0][value]= string
```

Response**General structure**

```
object {
entryid int //New glossary entry ID
warnings Optional //List of warnings
list of (
    //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [entryid] => int
    [warnings] =>
        Array
        (
            [0] =>
                Array
                (
                    [item] => string
                    [itemid] => int
                    [warningcode] => string
                    [message] => string
                )
        )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="entryid">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_glossary_delete_entry

Delete the given entry from the glossary.

Arguments

`entryid` (Required)

Glossary entry id to delete

General structure

```
int //Glossary entry id to delete
```

XML-RPC (PHP structure)

```
[entryid] => int
```

REST (POST parameters)

```
entryid= int
```

Response

General structure

```
object {
  result int //The processing result
  warnings Optional //list of warnings
  list of (
    //warning
  object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
  }
)}
```

XML-RPC (PHP structure)

```
Array
(
  [result] => int
  [warnings] =>
    Array
      (
        [0] =>
          Array
            (
              [item] => string
              [itemid] => int
              [warningcode] => string
              [message] => string
            )
      )
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="result">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod_glossary_get_authors**

Get the authors.

Arguments**id** (Required)

Glossary entry ID

General structure

int //Glossary entry ID

XML-RPC (PHP structure)

[id] => int

REST (POST parameters)

id= int

from (Default to "0")

Start returning records from here

General structureint **Default to "0"** //Start returning records from here**XML-RPC (PHP structure)**

[from] => int

REST (POST parameters)

from= int

limit (Default to "20")

Number of records to return

General structureint **Default to "20"** //Number of records to return**XML-RPC (PHP structure)**

[limit] => int

REST (POST parameters)

limit= int

options (Default to "Array ()")

An array of options

General structure

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes self even if all of their entries require approval. When true, also includes authors only having entries pending approval.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
  Array
    (
      [includenotapproved] => int
    )
```

REST (POST parameters)

options[includenotapproved]= int

Response

General structure

```
object {
count int //The total number of records.
authors list of (
object {
id int //The user ID
fullname string //The fullname
pictureurl string //The picture URL
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)}
```

XML-RPC (PHP structure)

```
Array
(
    [count] => int
    [authors] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [id] => int
                            [fullname] => string
                            [pictureurl] => string
                        )
            )
    [warnings] =>
        Array
            (
                [0] =>
                    Array
                        (
                            [item] => string
                            [itemid] => int
                            [warningcode] => string
                            [message] => string
                        )
            )
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="authors">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="fullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="pictureurl">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

Callable from AJAX

No

mod_glossary_get_categories

Get the categories.

Arguments

id (Required)

The glossary ID

General structure

```
int //The glossary ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

from (Default to "0")

Start returning records from here

General structure

```
int Default to "0" //Start returning records from here
```

XML-RPC (PHP structure)

```
[from] => int
```

REST (POST parameters)

```
from= int
```

limit (Default to "20")

Number of records to return

General structure

```
int Default to "20" //Number of records to return
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```



Response

General structure

```

object {
count int //The total number of records.
categories list of (
object {
id int //The category ID
glossaryid int //The glossary ID
name string //The name of the category
usedynalink int //Whether the category is automatically linked
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)
}

```

XML-RPC (PHP structure)

```

Array
(
    [count] => int
    [categories] =>
        Array
        (
            [
            [0] =>
                Array
                (
                    [
                        [id] => int
                        [glossaryid] => int
                        [name] => string
                        [usedynalink] => int
                    )
                )
            )
        )
    [warnings] =>
        Array
        (
            [
            [0] =>
                Array
                (
                    [
                        [item] => string
                        [itemid] => int
                        [warningcode] => string
                        [message] => string
                    )
                )
            )
        )
)
)

```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="count">
            <VALUE>int</VALUE>
        </KEY>
        <KEY name="categories">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="glossaryid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="usedynalink">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
    <MESSAGE>Invalid parameter value detected</MESSAGE>
    <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod_glossary_get_entries_by_author

Browse entries by author.

Arguments

id (Required)

Glossary entry ID

General structure

```
int //Glossary entry ID
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

letter (Required)

First letter of firstname or lastname, or either keywords: 'ALL' or 'SPECIAL'.

General structure

```
string //First letter of firstname or Lastname, or either keywords: 'ALL' or 'SPECIAL'.
```

XML-RPC (PHP structure)

```
[letter] => string
```

REST (POST parameters)

```
letter= string
```

field (Default to "LASTNAME")

Search and order using: 'FIRSTNAME' or 'LASTNAME'

General structure

```
string Default to "LASTNAME" //Search and order using: 'FIRSTNAME' or 'LASTNAME'
```

XML-RPC (PHP structure)

```
[field] => string
```

**REST (POST parameters)**

```
field= string
```

sort (Default to "ASC")

The direction of the order: 'ASC' or 'DESC'

General structure

```
string Default to "ASC" //The direction of the order: 'ASC' or 'DESC'
```

XML-RPC (PHP structure)

```
[sort] => string
```

REST (POST parameters)

```
sort= string
```

from (Default to "0")

Start returning records from here

General structure

```
int Default to "0" //Start returning records from here
```

XML-RPC (PHP structure)

```
[from] => int
```

REST (POST parameters)

```
from= int
```

limit (Default to "20")

Number of records to return

General structure

```
int Default to "20" //Number of records to return
```

XML-RPC (PHP structure)

```
[limit] => int
```

REST (POST parameters)

```
limit= int
```

options (Default to "Array ()")

An array of options

General structure

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
  Array
    (
      [includenotapproved] => int
    )
```

REST (POST parameters)

```
options[includenotapproved]= int
```

Response

General structure

```

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
    value int   //Scale value/option id.
    name string //Scale name.
}
})
)
)ratings Optional //The ratings
list of (
object {
    itemid int //Item id.
    scaleid int Optional //Scale id.
    userid int Optional //User who rated id.
    aggregate double Optional //Aggregated ratings grade.
    aggregatestr string Optional //Aggregated ratings as string.
    aggregatelabel string Optional //The aggregation label.
    count int Optional //Ratings count (used when aggregating).
    rating int Optional //The rating the user gave.
    canrate int Optional //Whether the user can rate the item.
    canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
)
warnings Optional //List of warnings
list of (
    //warning
object {
    item string Optional //item
    itemid int Optional //item id
    warningcode string //the warning code can be used by the client app to implement specific behaviour
    message string //untranslated english message to explain the warning
}
)
)
```



XML-RPC (PHP structure)

```

        )
    [count] => int
    [entries] =>
        Array
            (
                [
                    [0] =>
                        Array
                            (
                                [
                                    [id] => int
                                    [glossaryid] => int
                                    [userid] => int
                                    [userfullname] => string
                                    [userpictureurl] => string
                                    [concept] => string
                                    [definition] => string
                                    [definitionformat] => int
                                    [definitiontrust] => int
                                    [definitioninlinefiles] =>
                                        Array
                                            (
                                                [
                                                    [0] =>
                                                        Array
                                                            (
                                                                [
                                                                    [filename] => string
                                                                    [filepath] => string
                                                                    [filesize] => int
                                                                    [fileurl] => string
                                                                    [timemodified] => int
                                                                    [mimetype] => string
                                                                    [isexternalfile] => int
                                                                    [repositorytype] => string
                                                                )
                                                            )
                                                        )
                                                )
                                            )
                                        )
                                    [attachment] => int
                                    [attachments] =>
                                        Array
                                            (
                                                [
                                                    [0] =>
                                                        Array
                                                            (
                                                                [
                                                                    [filename] => string
                                                                    [filepath] => string
                                                                    [filesize] => int
                                                                    [fileurl] => string
                                                                    [timemodified] => int
                                                                    [mimetype] => string
                                                                    [isexternalfile] => int
                                                                    [repositorytype] => string
                                                                )
                                                            )
                                                        )
                                                )
                                            )
                                        )
                                    [timecreated] => int
                                    [timemodified] => int
                                    [teacherentry] => int
                                    [sourceglossaryid] => int
                                    [usedynalink] => int
                                    [casesensitive] => int
                                    [fullmatch] => int
                                    [approved] => int
                                    [tags] =>
                                        Array
                                            (
                                                [
                                                    [0] =>
                                                        Array
                                                            (
                                                                [
                                                                    [id] => int
                                                                    [name] => string
                                                                    [rawname] => string
                                                                    [isstandard] => int
                                                                    [tagcollid] => int
                                                                    [taginstanceid] => int
                                                                    [taginstancecontextid] => int
                                                                    [itemid] => int
                                                                    [ordering] => int
                                                                    [flag] => int
                                                                )
                                                            )
                                                        )
                                                )
                                            )
                                        )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )
)
```

```



```
Array
(
 [
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [value] => int
 [name] => string
)
)
)
)
)
)
)
 [ratings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entries\_by\_author\_id

Browse entries by author ID.



### Arguments

id (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

authorid (Required)

The author ID

**General structure**

```
int //The author ID
```

**XML-RPC (PHP structure)**

```
[authorid] => int
```

**REST (POST parameters)**

```
authorid= int
```

**order** (Default to "CONCEPT")

Order by: 'CONCEPT', 'CREATION' or 'UPDATE'

**General structure**

```
string Default to "CONCEPT" //Order by: 'CONCEPT', 'CREATION' or 'UPDATE'
```

**XML-RPC (PHP structure)**

```
[order] => string
```

**REST (POST parameters)**

```
order= string
```

**sort** (Default to "ASC")

The direction of the order: 'ASC' or 'DESC'

**General structure**

```
string Default to "ASC" //The direction of the order: 'ASC' or 'DESC'
```

**XML-RPC (PHP structure)**

```
[sort] => string
```

**REST (POST parameters)**

```
sort= string
```



**from** (Default to "0")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```

**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```

**options**(Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the
user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**

```
options[includenotapproved]= int
```

Response



```

General structure

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
})
)
)ratings Optional //The ratings
list of (
object {
itemid int //Item id.
scaleid int Optional //Scale id.
userid int Optional //User who rated id.
aggregate double Optional //Aggregated ratings grade.
aggregatestr string Optional //Aggregated ratings as string.
aggregatelabel string Optional //The aggregation Label.
count int Optional //Ratings count (used when aggregating).
rating int Optional //The rating the user gave.
canrate int Optional //Whether the user can rate the item.
canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
)
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [value] => int
 [name] => string
)
)
)
)
)
)
)
 [ratings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entries\_by\_category

Browse entries by category.



### Arguments

**id** (Required)

The glossary ID.

#### General structure

```
int //The glossary ID.
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

categoryid (Required)

The category ID. Use '0' for all categories, or '-1' for uncategorised entries.

**General structure**

```
int //The category ID. Use '0' for all categories, or '-1' for uncategorised entries.
```

**XML-RPC (PHP structure)**

```
[categoryid] => int
```

**REST (POST parameters)**

```
categoryid= int
```

**from** (Default to "0")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```

**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```



**options** (Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the
user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**

```
options[includenotapproved]= int
```

**Response**



```
items Optional //Scale items. Only returned for not numerical scales.
list of (
object {
value int //Scale value/option id.
name string //Scale name.
}
)
)
ratings Optional //The ratings
list of (
object {
itemid int //Item id.
scaleid int Optional //Scale id.
userid int Optional //User who rated id.
aggregate double Optional //Aggregated ratings grade.
aggregatestr string Optional //Aggregated ratings as string.
aggregatelabel string Optional //The aggregation label.
count int Optional //Ratings count (used when aggregating).
rating int Optional //The rating the user gave.
canrate int Optional //Whether the user can rate the item.
canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
}
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
 [categoryid] => int
 [categoryname] => string
)
)
)
)

```



```
)
 [ratinginfo] =>
 Array
 (
 [
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [value] => int
 [name] => string
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
)
[ratings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
)
)
)
)
)
[warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="categoryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="categoryname">
 <VALUE>string</VALUE>
```



```
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratinginfo">
<SINGLE>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="component">
<VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
<VALUE>string</VALUE>
</KEY>
<KEY name="canviewall">
<VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
<VALUE>int</VALUE>
</KEY>
<KEY name="scales">
<MULTIPLE>
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="max">
<VALUE>int</VALUE>
</KEY>
<KEY name="isnumeric">
<VALUE>int</VALUE>
</KEY>
<KEY name="items">
<MULTIPLE>
<SINGLE>
<KEY name="value">
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratings">
<MULTIPLE>
<SINGLE>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
<VALUE>int</VALUE>
</KEY>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="aggregate">
<VALUE>double</VALUE>
</KEY>
<KEY name="aggregatestr">
<VALUE>string</VALUE>
</KEY>
<KEY name="aggregatelabel">
<VALUE>string</VALUE>
</KEY>
<KEY name="count">
<VALUE>int</VALUE>
</KEY>
<KEY name="rating">
<VALUE>int</VALUE>
</KEY>
<KEY name="canrate">
<VALUE>int</VALUE>
```

^

```

 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 <KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No



### mod\_glossary\_get\_entries\_by\_date

Browse entries by date.

### Arguments

**id** (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

**REST (POST parameters)**

id= int

**order** (Default to "UPDATE")

Order the records by: 'CREATION' or 'UPDATE'.

**General structure**`string Default to "UPDATE" //Order the records by: 'CREATION' or 'UPDATE'.`**XML-RPC (PHP structure)**`[order] => string`**REST (POST parameters)**

order= string

**sort** (Default to "DESC")

The direction of the order: 'ASC' or 'DESC'

**General structure**`string Default to "DESC" //The direction of the order: 'ASC' or 'DESC'`**XML-RPC (PHP structure)**`[sort] => string`**REST (POST parameters)**

sort= string

**from** (Default to "0")

Start returning records from here

**General structure**`int Default to "0" //Start returning records from here`**XML-RPC (PHP structure)**`[from] => int`**REST (POST parameters)**

from= int

**limit** (Default to "20")

Number of records to return

**General structure**`int Default to "20" //Number of records to return`**XML-RPC (PHP structure)**`[limit] => int`

**REST (POST parameters)**`limit= int``options (Default to "Array ()")`

An array of options

**General structure**

```
Default to "Array
(
)
"
 //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the
user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**`options[includenotapproved]= int`

Response



**General structure**

```

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
})
)
)ratings Optional //The ratings
list of (
object {
itemid int //Item id.
scaleid int Optional //Scale id.
userid int Optional //User who rated id.
aggregate double Optional //Aggregated ratings grade.
aggregatestr string Optional //Aggregated ratings as string.
aggregatelabel string Optional //The aggregation label.
count int Optional //Ratings count (used when aggregating).
rating int Optional //The rating the user gave.
canrate int Optional //Whether the user can rate the item.
canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
)
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)
```



## XML-RPC (PHP structure)

```

 (
 [count] => int
 [entries] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
]
 [attachment] => int
 [attachments] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
]
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
]
)
)
)
)
)
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [name] => string
)
)
)
)
 [ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entries\_by\_letter

Browse entries by letter.



### Arguments

id (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

letter (Required)

A letter, or either keywords: 'ALL' or 'SPECIAL'.

**General structure**

```
string //A Letter, or either keywords: 'ALL' or 'SPECIAL'.
```

**XML-RPC (PHP structure)**

```
[letter] => string
```

**REST (POST parameters)**

```
letter= string
```

**from** (Default to "0")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```

**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```



**options** (Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the
user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**

```
options[includenotapproved]= int
```

**Response**

```

General structure

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
)}
)ratings Optional //The ratings
list of (
object {
 itemid int //Item id.
 scaleid int Optional //Scale id.
 userid int Optional //User who rated id.
 aggregate double Optional //Aggregated ratings grade.
 aggregatestr string Optional //Aggregated ratings as string.
 aggregatelabel string Optional //The aggregation Label.
 count int Optional //Ratings count (used when aggregating).
 rating int Optional //The rating the user gave.
 canrate int Optional //Whether the user can rate the item.
 canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
warnings Optional //List of warnings
list of (
 //warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [name] => string
)
)
)
)
)
[ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entries\_by\_search

Browse entries by search query.



### Arguments

**id** (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

**query** (Required)

The query string

**General structure**

```
string //The query string
```

**XML-RPC (PHP structure)**

```
[query] => string
```

**REST (POST parameters)**

```
query= string
```

**fullsearch** (Default to "1")

The query

**General structure**

```
int Default to "1" //The query
```

**XML-RPC (PHP structure)**

```
[fullsearch] => int
```

**REST (POST parameters)**

```
fullsearch= int
```

**order** (Default to "CONCEPT")

Order by: 'CONCEPT', 'CREATION' or 'UPDATE'

**General structure**

```
string Default to "CONCEPT" //Order by: 'CONCEPT', 'CREATION' or 'UPDATE'
```

**XML-RPC (PHP structure)**

```
[order] => string
```

**REST (POST parameters)**

```
order= string
```



**sort** (Default to "ASC")

The direction of the order: 'ASC' or 'DESC'

**General structure**

```
string Default to "ASC" //The direction of the order: 'ASC' or 'DESC'
```

**XML-RPC (PHP structure)**

```
[sort] => string
```

**REST (POST parameters)**

```
sort= string
```

**from** (Default to "o")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```

**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```

**options** (Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**

```
options[includenotapproved]= int
```

**Response**

**General structure**

```

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
)}
)ratings Optional //The ratings
list of (
object {
 itemid int //Item id.
 scaleid int Optional //Scale id.
 userid int Optional //User who rated id.
 aggregate double Optional //Aggregated ratings grade.
 aggregatestr string Optional //Aggregated ratings as string.
 aggregatelabel string Optional //The aggregation Label.
 count int Optional //Ratings count (used when aggregating).
 rating int Optional //The rating the user gave.
 canrate int Optional //Whether the user can rate the item.
 canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
})
warnings Optional //List of warnings
list of (
 //warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [name] => string
)
)
)
)
 [ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entries\_by\_term

Browse entries by term (concept or alias).



### Arguments

**id** (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

**term** (Required)

The entry concept, or alias

**General structure**

```
string //The entry concept, or alias
```

**XML-RPC (PHP structure)**

```
[term] => string
```

**REST (POST parameters)**

```
term= string
```

**from** (Default to "0")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```

**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```



**options** (Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
includenotapproved int Default to "0" //When false, includes the non-approved entries created by the user. When true, also includes the ones that the
user has the permission to approve.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [includenotapproved] => int
)
```

**REST (POST parameters)**

```
options[includenotapproved]= int
```

**Response**

**General structure**

```

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
)}
)ratings Optional //The ratings
list of (
object {
 itemid int //Item id.
 scaleid int Optional //Scale id.
 userid int Optional //User who rated id.
 aggregate double Optional //Aggregated ratings grade.
 aggregatestr string Optional //Aggregated ratings as string.
 aggregatelabel string Optional //The aggregation Label.
 count int Optional //Ratings count (used when aggregating).
 rating int Optional //The rating the user gave.
 canrate int Optional //Whether the user can rate the item.
 canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
})
warnings Optional //List of warnings
list of (
 //warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [name] => string
)
)
)
)
)
[ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

[mod\\_glossary\\_get\\_entries\\_to\\_approve](#)

Browse entries to be approved.



### Arguments

id (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

letter (Required)

A letter, or either keywords: 'ALL' or 'SPECIAL'.

**General structure**

```
string //A Letter, or either keywords: 'ALL' or 'SPECIAL'.
```

**XML-RPC (PHP structure)**

```
[letter] => string
```

**REST (POST parameters)**

```
letter= string
```

**order** (Default to "CONCEPT")

Order by: 'CONCEPT', 'CREATION' or 'UPDATE'

**General structure**

```
string Default to "CONCEPT" //Order by: 'CONCEPT', 'CREATION' or 'UPDATE'
```

**XML-RPC (PHP structure)**

```
[order] => string
```

**REST (POST parameters)**

```
order= string
```

**sort** (Default to "ASC")

The direction of the order: 'ASC' or 'DESC'

**General structure**

```
string Default to "ASC" //The direction of the order: 'ASC' or 'DESC'
```

**XML-RPC (PHP structure)**

```
[sort] => string
```

**REST (POST parameters)**

```
sort= string
```



**from** (Default to "0")

Start returning records from here

**General structure**

```
int Default to "0" //Start returning records from here
```

**XML-RPC (PHP structure)**

```
[from] => int
```

**REST (POST parameters)**

```
from= int
```



**limit** (Default to "20")

Number of records to return

**General structure**

```
int Default to "20" //Number of records to return
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```

**options**(Default to "Array ()")

An array of options

**General structure**

```
Default to "Array
(
)
" //An array of options
object {
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
)
```

**REST (POST parameters)**

**Response**



```

General structure

object {
count int //The total number of records matching the request.
entries list of (
object {
id int //The entry ID
glossaryid int //The glossary ID
userid int //Author ID
userfullname string //Author full name
userpictureurl string //Author picture
concept string //The concept
definition string //The definition
definitionformat int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
definitiontrust int //The definition trust flag
definitioninlinefiles Optional //entry definition inline files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)attachment int //Whether or not the entry has attachments
attachments Optional //attachments
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
timecreated int //Time created
timemodified int //Time modified
teacherentry int //The entry was created by a teacher, or equivalent.
sourceglossaryid int //The source glossary ID
usedynalink int //Whether the concept should be automatically Linked
casesensitive int //When true, the matching is case sensitive
fullmatch int //When true, the matching is done on full words only
approved int //Whether the entry was approved
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
}
)
)ratinginfo Optional //Rating information
object {
contextid int //Context id.
component string //Context name.
ratingarea string //Rating area name.
canviewall int Optional //Whether the user can view all the individual ratings.
canviewany int Optional //Whether the user can view aggregate of ratings of others.
scales Optional //Different scales used information
list of (
//Scale information
object {
id int //Scale id.
courseid int Optional //Course id.
name string Optional //Scale name (when a real scale is used).
max int //Max value for the scale.
isnumeric int //Whether is a numeric scale.
items Optional //Scale items. Only returned for not numerical scales.
list of (

```



```
object {
 value int //Scale value/option id.
 name string //Scale name.
}
)}
)ratings Optional //The ratings
list of (
object {
 itemid int //Item id.
 scaleid int Optional //Scale id.
 userid int Optional //User who rated id.
 aggregate double Optional //Aggregated ratings grade.
 aggregatestr string Optional //Aggregated ratings as string.
 aggregatelabel string Optional //The aggregation Label.
 count int Optional //Ratings count (used when aggregating).
 rating int Optional //The rating the user gave.
 canrate int Optional //Whether the user can rate the item.
 canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
)
})
warnings Optional //List of warnings
list of (
 //warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [count] => int
 [entries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
)
[ratinginfo] =>

```



```
Array
(
 [contextid] => int
 [component] => string
 [ratingarea] => string
 [canviewall] => int
 [canviewany] => int
 [scales] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [name] => string
)
)
)
)
 [ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="entries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
```



```
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="component">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_get\_entry\_by\_id

Get an entry by ID



### Arguments

id (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

## Response

^



```
name string //Scale name.
}
}
)
)
)
ratings Optional //The ratings
list of (
object {
itemid int //Item id.
scaleid int Optional //Scale id.
userid int Optional //User who rated id.
aggregate double Optional //Aggregated ratings grade.
aggregatestr string Optional //Aggregated ratings as string.
aggregatelabel string Optional //The aggregation label.
count int Optional //Ratings count (used when aggregating).
rating int Optional //The rating the user gave.
canrate int Optional //Whether the user can rate the item.
canviewaggregate int Optional //Whether the user can view the aggregated grade.
}
}
}
permissions Optional //User permissions for the managing the entry.
object {
candelete int //Whether the user can delete the entry.
canupdate int //Whether the user can update the entry.
}
}
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
}
})
```



**XML-RPC (PHP structure)**

```

Array
(
 [entry] =>
 Array
 (
 [id] => int
 [glossaryid] => int
 [userid] => int
 [userfullname] => string
 [userpictureurl] => string
 [concept] => string
 [definition] => string
 [definitionformat] => int
 [definitiontrust] => int
 [definitioninlinefiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [attachment] => int
 [attachments] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [timecreated] => int
 [timemodified] => int
 [teacherentry] => int
 [sourceglossaryid] => int
 [usedynalink] => int
 [casesensitive] => int
 [fullmatch] => int
 [approved] => int
 [tags] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
 [ratinginfo] =>
 Array
 (
 [
 [contextid] => int
 [component] => string
 [ratingarea] => string
]
)
)
)

```



```
[canviewall] => int
[canviewany] => int
[scales] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [courseid] => int
 [name] => string
 [max] => int
 [isnumeric] => int
 [items] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [value] => int
 [name] => string
)
)
)
)
)
)
)
)
[ratings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [itemid] => int
 [scaleid] => int
 [userid] => int
 [aggregate] => double
 [aggregatestr] => string
 [aggregatelabel] => string
 [count] => int
 [rating] => int
 [canrate] => int
 [canviewaggregate] => int
)
)
)
)
)
[permissions] =>
 Array
 (
 [
 [candelete] => int
 [canupdate] => int
]
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="entry">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="glossaryid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userfullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userpictureurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="concept">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definition">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="definitionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitiontrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="definitioninlinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
```

```

 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="teacherentry">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sourceglossaryid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="casesensitive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullmatch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approved">
 <VALUE>int</VALUE>
</KEY>
<KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="ratinginfo">
 <SINGLE>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>

```



```
</KEY>
<KEY name="ratingarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canviewwall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canviewany">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scales">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="max">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isnumeric">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="items">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="ratings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="aggregate">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="aggregatestr">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="aggregatelabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rating">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewaggregate">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="permissions">
 <SINGLE>
```



```

<KEY name="candelete">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canupdate">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No



[mod\\_glossary\\_get\\_glossaries\\_by\\_courses](#)

Retrieve a list of glossaries from several courses.

### Arguments

courseids (Default to "Array ()")

Array of course IDs

#### General structure

```

Default to "Array
(
)
" //Array of course IDs
list of (
int //course id
)

```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [
 [0] => int
)
)
```

**REST (POST parameters)**

```
courseids[0]= int
```

**Response**

**General structure**

```

object {
glossaries list of (
 //Glossaries
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
allowduplicatedentries int //If enabled, multiple entries can have the same concept name
displayformat string //Display format type
mainglossary int //If enabled this glossary is a main glossary.
showspecial int //If enabled, participants can browse the glossary by special characters, such as @ and #
showalphabet int //If enabled, participants can browse the glossary by letters of the alphabet
showall int //If enabled, participants can browse all entries at once
allowcomments int //If enabled, all participants with permission to create comments will be able to add comments to glossary entries
allowprintview int //If enabled, students are provided with a link to a printer-friendly version of the glossary. The link is always available to teachers
usedynalink int //If site-wide glossary auto-linking has been enabled by an administrator and this checkbox is ticked, the entry will be automatically linked wherever the concept words and phrases appear throughout the rest of the course.
defaultapproval int //If set to no, entries require approving by a teacher before they are viewable by everyone.
approvaldisplayformat string //When approving glossary items you may wish to use a different display format
globalglossary int entbypage int //Entries shown per page
editalways int //Always allow editing
rssstype int //To enable the RSS feed for this activity, select either concepts with author or concepts without author to be included in the feed
rssarticles int //This setting specifies the number of glossary entry concepts to include in the RSS feed. Between 5 and 20 generally acceptable
assessed int //Aggregate type
assesstimestart int //Restrict rating to items created after this
assesstimefinish int //Restrict rating to items created before this
scale int //Scale ID
timecreated int //Time created
timemodified int //Time modified
completionentries int //Number of entries to complete
browsemode list of (
string //Modes of browsing allowed
)canaddentry int Optional //Whether the user can add a new entry
)
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [glossaries] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [allowduplicatedentries] => int
 [displayformat] => string
 [mainglossary] => int
 [showspecial] => int
 [showalphabet] => int
 [showall] => int
 [allowcomments] => int
 [allowprintview] => int
 [usedynalink] => int
 [defaultapproval] => int
 [approvaldisplayformat] => string
 [globalglossary] => int
 [entbypage] => int
 [editalways] => int
 [rsstype] => int
 [rssarticles] => int
 [assessed] => int
 [assesstimestart] => int
 [assesstimefinish] => int
 [scale] => int
 [timecreated] => int
 [timemodified] => int
 [completionentries] => int
 [browsemode] =>
 Array
 (
 [0] => string
)
 [canaddentry] => int
)
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```





^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="glossaries">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="allowduplicatedentries">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayformat">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="mainglossary">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showspecial">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="showalphabet">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showall">
 <VALUE>int</VALUE>
</KEY>
<KEY name="allowcomments">
 <VALUE>int</VALUE>
</KEY>
<KEY name="allowprintview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usedynalink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="defaultapproval">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approvaldisplayformat">
 <VALUE>string</VALUE>
</KEY>
<KEY name="globalglossary">
 <VALUE>int</VALUE>
</KEY>
<KEY name="entbypage">
 <VALUE>int</VALUE>
</KEY>
<KEY name="editalways">
 <VALUE>int</VALUE>
</KEY>
<KEY name="rssstype">
 <VALUE>int</VALUE>
</KEY>
<KEY name="rssarticles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="assessed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="assesstimestart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="assesstimefinish">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scale">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completionentries">
 <VALUE>int</VALUE>
</KEY>
<KEY name="browsemode">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
</KEY>
<KEY name="canaddentry">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```

### Error message

#### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_glossary\_prepare\_entry\_for\_edition

Prepares the given entry for edition returning draft item areas and file areas information.

### Arguments

#### entryid (Required)

Glossary entry id to update

#### General structure

```
int //Glossary entry id to update
```

#### XML-RPC (PHP structure)

```
[entryid] => int
```



#### REST (POST parameters)

```
entryid= int
```

### Response

**General structure**

```
object {
 inlineattachmentsid int //Draft item id for the text editor.
 attachmentsid int //Draft item id for the file manager.
 areas //File areas including options
 list of (
 object {
 area string //File area name.
 options //Draft file area options.
 list of (
 object {
 name string //Name of option.
 value string //Value of option.
 }
)
 }
)
 aliases list of (
 string //Alias name.
)
 categories list of (
 int //Category id
)
 warnings Optional //List of warnings
 list of (
 //warning
)
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [inlineattachmentsid] => int
 [attachmentsid] => int
 [areas] =>
 Array
 (
 [0] =>
 Array
 (
 [area] => string
 [options] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
)
)
 [aliases] =>
 Array
 (
 [0] => string
)
 [categories] =>
 Array
 (
 [0] => int
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="inlineattachmentsid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachmentsid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="areas">
 <MULTIPLE>
 <SINGLE>
 <KEY name="area">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="aliases">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="categories">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_glossary_update_entry`

Updates the given glossary entry.

#### Arguments

`entryid` (Required)

Glossary entry id to update

**General structure**

```
int //Glossary entry id to update
```

**XML-RPC (PHP structure)**

```
[entryid] => int
```

**REST (POST parameters)**

```
entryid= int
```

`concept` (Required)

Glossary concept

**General structure**

```
string //Glossary concept
```

**XML-RPC (PHP structure)**

```
[concept] => string
```

**REST (POST parameters)**

```
concept= string
```



`definition` (Required)

Glossary concept definition

**General structure**

```
string //Glossary concept definition
```

**XML-RPC (PHP structure)**

```
[definition] => string
```

**REST (POST parameters)**

```
definition= string
```

`definitionformat` (Required)

definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN

**General structure**

```
int //definition format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
```

**XML-RPC (PHP structure)**

```
[definitionformat] => int
```

**REST (POST parameters)**

```
definitionformat= int
```

**options** (Default to "Array ()")

Optional settings

**General structure**

```
Default to "Array
(
)
" //Optional settings
list of (
object {
name string //The allowed keys (value format) are:
 inlineattachmentsid (int); the draft file area id for inline attachments
 attachmentsid (int); the draft file area id for attachments
 categories (comma separated int); comma separated category ids
 aliases (comma separated str); comma separated aliases
 usedynalink (bool); whether the entry should be automatically Linked.
 casesensitive (bool); whether the entry is case sensitive.
 fullmatch (bool); whether to match whole words only.
value string //the value of the option (validated inside the function)
}
)
```

**XML-RPC (PHP structure)**

```
[options] =>
Array
(
[0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
```

**REST (POST parameters)**

```
options[0][name]= string
options[0][value]= string
```

**Response**

**General structure**

```

object {
 result int //The update result
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod\_glossary\_view\_entry

Notify a glossary entry as being viewed.

#### Arguments

**id** (Required)

Glossary entry ID

#### General structure

```
int //Glossary entry ID
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

#### Response

#### General structure

```
object {
 status int //True on success
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



#### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_glossary\_view\_glossary**

Notify the glossary as being viewed.

**Arguments****id** (Required)

Glossary instance ID

**General structure**

```
int //Glossary instance ID
```

**XML-RPC (PHP structure)**

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

**mode (Required)**

The mode in which the glossary is viewed

**General structure**

```
string //The mode in which the glossary is viewed
```

**XML-RPC (PHP structure)**

```
[mode] => string
```

**REST (POST parameters)**

```
mode= string
```

**Response****General structure**

```
object {
 status int //True on success
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_h5pactivity\_get\_attempts**

Return the information needed to list a user attempts.

**Arguments****h5pactivityid** (Required)

h5p activity instance id

**General structure**

```
int //h5p activity instance id
```

**XML-RPC (PHP structure)**

```
[h5pactivityid] => int
```

**REST (POST parameters)**`h5pactivityid= int``userids (Default to "Array ()")`

User ids

**General structure**

```
Default to "Array
(
)
" //User ids
list of (
int Default to "null" //The user ids to get attempts (null means only current user)
)
```

**XML-RPC (PHP structure)**

```
[userids] =>
 Array
 (
 [0] => int
)
```

**REST (POST parameters)**`userids[0]= int`**Response**

Activity attempts data



**General structure**

```
//Activity attempts data
object {
activityid int //Activity course module ID
usersattempts //The complete users attempts list
list of (
object {
userid int //The user id
attempts //The complete attempts list
list of (
object {
id int //ID of the context
h5activityid int //ID of the H5P activity
userid int //ID of the user
timecreated int //Attempt creation
timemodified int //Attempt modified
attempt int //Attempt number
rawscore int //Attempt score value
maxscore int //Attempt max score
duration int //Attempt duration in seconds
completion int Optional //Attempt completion
success int Optional //Attempt success
scaled double //Attempt scaled
}
)scored Optional //Attempts used to grade the activity
object {
title string //Scored attempts title
grademethod string //Scored attempts title
attempts //List of the grading attempts
list of (
object {
id int //ID of the context
h5activityid int //ID of the H5P activity
userid int //ID of the user
timecreated int //Attempt creation
timemodified int //Attempt modified
attempt int //Attempt number
rawscore int //Attempt score value
maxscore int //Attempt max score
duration int //Attempt duration in seconds
completion int Optional //Attempt completion
success int Optional //Attempt success
scaled double //Attempt scaled
}
)
)
)
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [activityid] => int
 [usersattempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [attempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [h5activityid] => int
 [userid] => int
 [timecreated] => int
 [timemodified] => int
 [attempt] => int
 [rawscore] => int
 [maxscore] => int
 [duration] => int
 [completion] => int
 [success] => int
 [scaled] => double
)
)
)
)
)
 [scored] =>
 Array
 (
 [
 [title] => string
 [grademethod] => string
 [attempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [h5activityid] => int
 [userid] => int
 [timecreated] => int
 [timemodified] => int
 [attempt] => int
 [rawscore] => int
 [maxscore] => int
 [duration] => int
 [completion] => int
 [success] => int
 [scaled] => double
)
)
)
)
)
)
)
)
)
)
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)

```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="activityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usersattempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="h5activityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rawscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duration">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completion">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaled">
 <VALUE>double</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="scored">
 <SINGLE>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="grademethod">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="h5activityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">

```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="rawscore">
 <VALUE>int</VALUE>
</KEY>
<KEY name="maxscore">
 <VALUE>int</VALUE>
</KEY>
<KEY name="duration">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completion">
 <VALUE>int</VALUE>
</KEY>
<KEY name="success">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaled">
 <VALUE>double</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_h5pactivity\_get\_h5pactivities\_by\_courses

Returns a list of h5p activities in a list of provided courses, if no list is provided all h5p activities that the user can view will be returned.

## Arguments

courseids (Default to "Array ()")

Array of course ids

### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)
```

### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

### REST (POST parameters)

```
courseids[0]= int
```

## Response



**General structure**

```

object {
 h5pactivities list of (
 object {
 id int //The primary key of the record.
 course int //Course id this h5p activity is part of.
 name string //The name of the activity module instance.
 timecreated int Optional //Timestamp of when the instance was added to the course.
 timemodified int Optional //Timestamp of when the instance was last modified.
 intro string //H5P activity description.
 introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 grade int Optional //The maximum grade for submission.
 displayoptions int //H5P Button display options.
 enabletracking int //Enable xAPI tracking.
 grademethod int //Which H5P attempt is used for grading.
 contenthash string Optional //Sha1 hash of file content.
 coursemodule int //coursemodule
 context int //context
 introfiles //introfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)package //package
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)deployedfile object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 }
)
)h5pglobalsettings Optional //H5P global settings
 object {
 enablesavestate int //Whether saving state is enabled.
 savestatefreq int Optional //How often (in seconds) state is saved.
 }
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
)

```



**XML-RPC (PHP structure)**

```

Array
(
 [h5pactivities] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [course] => int
 [name] => string
 [timecreated] => int
 [timemodified] => int
 [intro] => string
 [introformat] => int
 [grade] => int
 [displayoptions] => int
 [enabletracking] => int
 [grademethod] => int
 [contenthash] => string
 [coursemodule] => int
 [context] => int
 [introfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [package] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [deployedfile] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
)
)
)
)
)
 [h5pglobalssettings] =>
 Array
 (
 [enablesavestate] => int
 [savestatefreq] => int
)
)
[warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
)
)
)
)
)

```



```
[warningcode] => string
[message] => string
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="h5pactivities">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayoptions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enabletracking">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grademethod">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contenthash">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="context">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="package">
 <MULTIPLE>
 <SINGLE>
```



```
<KEY name="filename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="deployedfile">
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="h5pglobalsettings">
 <SINGLE>
 <KEY name="enablesavestate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="savestatefreq">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



## Error message

### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

No

## mod\_h5pactivity\_get\_h5pactivity\_access\_information

Return access information for a given h5p activity.

## Arguments

**h5pactivityid** (Required)

h5p activity instance id

### General structure

```
int //h5p activity instance id
```

### XML-RPC (PHP structure)

```
[h5pactivityid] => int
```

### REST (POST parameters)

```
h5pactivityid= int
```

## Response

### General structure

```
object {
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)canview int Optional //Whether the user has the capability mod/h5pactivity:view allowed.
 canaddinstance int Optional //Whether the user has the capability mod/h5pactivity:addinstance allowed.
 cansubmit int Optional //Whether the user has the capability mod/h5pactivity:submit allowed.
 canreviewattempts int Optional //Whether the user has the capability mod/h5pactivity:reviewattempts allowed.
}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
]
)
 [canview] => int
 [canaddinstance] => int
 [cansubmit] => int
 [canreviewattempts] => int
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="canview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canaddinstance">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cansubmit">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreviewattempts">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

**Callable from AJAX**

No

**mod\_h5pactivity\_get\_results**

Return the information needed to list a user attempt results.

**Arguments****h5pactivityid** (Required)

h5p activity instance id

**General structure**

```
int //h5p activity instance id
```

**XML-RPC (PHP structure)**

```
[h5pactivityid] => int
```

**REST (POST parameters)**

```
h5pactivityid= int
```

**attemptids** (Default to "Array ()")

Attempt ids

**General structure**

```
Default to "Array
(
)
" //Attempt ids
list of (
int //The attempt id
)
```

**XML-RPC (PHP structure)**

```
[attemptids] =>
 Array
 (
 [0] => int
)

```

**REST (POST parameters)**

```
attemptids[0]= int
```

**Response**

Activity attempts results data

^

**General structure**

```

//Activity attempts results data
object {
activityid int //Activity course module ID
attempts //The complete attempts list
list of (
 //The attempt general information
object {
id int //ID of the context
h5activityid int //ID of the H5P activity
userid int //ID of the user
timecreated int //Attempt creation
timemodified int //Attempt modified
attempt int //Attempt number
rawscore int //Attempt score value
maxscore int //Attempt max score
duration int //Attempt duration in seconds
completion int Optional //Attempt completion
success int Optional //Attempt success
scaled double //Attempt scaled
results Optional //The results of the attempt
list of (
 //A single result statement tracking information
object {
id int //ID of the context
attemptid int //ID of the H5P attempt
subcontent string //Subcontent identifier
timecreated int //Result creation
interactiontype string //Interaction type
description string //Result description
content string Optional //Result extra content
rawscore int //Result score value
maxscore int //Result max score
duration int Optional //Result duration in seconds
completion int Optional //Result completion
success int Optional //Result success
optionslabel string Optional //Label used for result options
correctlabel string Optional //Label used for correct answers
answerlabel string Optional //Label used for user answers
track int Optional //If the result has valid track information
options Optional //The statement options
list of (
object {
description string Optional //Option description
id string Optional //Option string identifier
correctanswer Optional //The option correct answer
object {
answer string Optional //Option text value
correct int Optional //If has to be displayed as correct
incorrect int Optional //If has to be displayed as incorrect
text int Optional //If has to be displayed as simple text
checked int Optional //If has to be displayed as a checked option
unchecked int Optional //If has to be displayed as a unchecked option
pass int Optional //If has to be displayed as passed
fail int Optional //If has to be displayed as failed
}
useranswer Optional //The option user answer
object {
answer string Optional //Option text value
correct int Optional //If has to be displayed as correct
incorrect int Optional //If has to be displayed as incorrect
text int Optional //If has to be displayed as simple text
checked int Optional //If has to be displayed as a checked option
unchecked int Optional //If has to be displayed as a unchecked option
pass int Optional //If has to be displayed as passed
fail int Optional //If has to be displayed as failed
}
}
)
})
)
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)

```





^

## XML-RPC (PHP structure)

```
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="activityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="h5pactivityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rawscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duration">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completion">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaled">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="results">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attemptid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="subcontent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="interactiontype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duration">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completion">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="success">
 <VALUE>int</VALUE>
</KEY>
<KEY name="optionslabel">
 <VALUE>string</VALUE>
</KEY>
<KEY name="correctlabel">
 <VALUE>string</VALUE>
</KEY>
<KEY name="answerlabel">
 <VALUE>string</VALUE>
</KEY>
<KEY name="track">
 <VALUE>int</VALUE>
</KEY>
<KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="correctanswer">
 <SINGLE>
 <KEY name="answer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="correct">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="incorrect">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="text">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="checked">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unchecked">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pass">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fail">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_h5pactivity\_get\_user\_attempts



Return the information needed to list all enrolled user attempts.

### Arguments

**h5pactivityid** (Required)

h5p activity instance id

#### General structure

```
int //h5p activity instance id
```

#### XML-RPC (PHP structure)

```
[h5pactivityid] => int
```

**REST (POST parameters)**

```
h5pactivityid= int
```

**sortorder** (Default to "id ASC")

sort by either user id, firstname or lastname (with optional asc/desc)

**General structure**

```
string Default to "id ASC" //sort by either user id, firstname or Lastname (with optional asc/desc)
```

**XML-RPC (PHP structure)**

```
[sortorder] => string
```

**REST (POST parameters)**

```
sortorder= string
```

**page** (Default to "-1")

current page

**General structure**

```
int Default to "-1" //current page
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

**perpage** (Default to "0")

items per page

**General structure**

```
int Default to "0" //items per page
```

**XML-RPC (PHP structure)**

```
[perpage] => int
```

**REST (POST parameters)**

```
perpage= int
```

**firstinitial** (Default to "")

Users whose first name starts with \$firstinitial

**General structure**

```
string Default to "" //Users whose first name starts with $firstinitial
```

**XML-RPC (PHP structure)**

```
[firstinitial] => string
```

**REST (POST parameters)**`firstinitial= string`**lastinitial (Default to "")**

Users whose last name starts with \$lastinitial

**General structure**`string Default to "" //Users whose Last name starts with $lastinitial`**XML-RPC (PHP structure)**`[lastinitial] => string`**REST (POST parameters)**`lastinitial= string`**Response**

Activity attempts data



**General structure**

```
//Activity attempts data
object {
activityid int //Activity course module ID
usersattempts //The complete users attempts list
list of (
object {
userid int //The user id
attempts //The complete attempts list
list of (
object {
id int //ID of the context
h5activityid int //ID of the H5P activity
userid int //ID of the user
timecreated int //Attempt creation
timemodified int //Attempt modified
attempt int //Attempt number
rawscore int //Attempt score value
maxscore int //Attempt max score
duration int //Attempt duration in seconds
completion int Optional //Attempt completion
success int Optional //Attempt success
scaled double //Attempt scaled
}
)scored Optional //Attempts used to grade the activity
object {
title string //Scored attempts title
grademethod string //Grading method
attempts //List of the grading attempts
list of (
object {
id int //ID of the context
h5activityid int //ID of the H5P activity
userid int //ID of the user
timecreated int //Attempt creation
timemodified int //Attempt modified
attempt int //Attempt number
rawscore int //Attempt score value
maxscore int //Attempt max score
duration int //Attempt duration in seconds
completion int Optional //Attempt completion
success int Optional //Attempt success
scaled double //Attempt scaled
}
)
)
)
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [activityid] => int
 [usersattempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [attempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [h5activityid] => int
 [userid] => int
 [timecreated] => int
 [timemodified] => int
 [attempt] => int
 [rawscore] => int
 [maxscore] => int
 [duration] => int
 [completion] => int
 [success] => int
 [scaled] => double
)
)
)
)
)
 [scored] =>
 Array
 (
 [
 [title] => string
 [grademethod] => string
 [attempts] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [h5activityid] => int
 [userid] => int
 [timecreated] => int
 [timemodified] => int
 [attempt] => int
 [rawscore] => int
 [maxscore] => int
 [duration] => int
 [completion] => int
 [success] => int
 [scaled] => double
)
)
)
)
)
)
)
)
)
)
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)

```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="activityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usersattempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="h5pactivityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rawscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxscore">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duration">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completion">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaled">
 <VALUE>double</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="scored">
 <SINGLE>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="grademethod">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="h5pactivityid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">

```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="rawscore">
 <VALUE>int</VALUE>
</KEY>
<KEY name="maxscore">
 <VALUE>int</VALUE>
</KEY>
<KEY name="duration">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completion">
 <VALUE>int</VALUE>
</KEY>
<KEY name="success">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaled">
 <VALUE>double</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

mod\_h5pactivity\_log\_report\_viewed

Log that the h5pactivity was viewed.

## Arguments

**h5pactivityid** (Required)

h5p activity instance id

### General structure

```
int //h5p activity instance id
```

### XML-RPC (PHP structure)

```
[h5pactivityid] => int
```

### REST (POST parameters)

```
h5pactivityid= int
```

**userid** (Default to "null")

The user id to log attempt (null means only current user)

### General structure

```
int Default to "null" //The user id to log attempt (null means only current user)
```

### XML-RPC (PHP structure)

```
[userid] => int
```

### REST (POST parameters)

```
userid= int
```

**attemptid** (Default to "null")

The attempt id

### General structure

```
int Default to "null" //The attempt id
```



### XML-RPC (PHP structure)

```
[attemptid] => int
```



### REST (POST parameters)

```
attemptid= int
```

## Response

**General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod\_h5pactivity\_view\_h5pactivity**

Trigger the course module viewed event and update the module completion status.

#### Arguments

**h5pactivityid** (Required)

H5P activity instance id

##### General structure

```
int //H5P activity instance id
```

##### XML-RPC (PHP structure)

```
[h5pactivityid] => int
```

##### REST (POST parameters)

```
h5pactivityid= int
```

#### Response

##### General structure

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



##### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_imscp\_get\_imscps\_by\_courses**

Returns a list of IMSCP instances in a provided set of courses, if no courses are provided then all the IMSCP instances the user has access to will be returned.

**Arguments**

courseids (Default to "Array ()")

Array of course ids

**General structure**

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [
 [0] => int
)
)
```

**REST (POST parameters)**

```
courseids[0]= int
```

**Response****General structure**

```
object {
 imscps list of (
 //IMS content packages
 object {
 id int //Activity instance id
 coursemodule int //Course module id
 course int //Course id
 name string //Activity name
 intro string Optional //Activity introduction
 introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 introfiles Optional //Files in the introduction
 list of (
 //File
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for external files.
 }
)section int Optional //Course section id
 visible int Optional //Visible
 groupmode int Optional //Group mode
 groupingid int Optional //Group id
 lang string Optional //Forced activity language
 revision int Optional //Revision
 keepold int Optional //Number of old IMSCP to keep
 structure string Optional //IMSCP structure
 timemodified string Optional //Time of last modification
)
)warnings Optional //List of warnings
list of (
 //warning
)object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
)
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [imscps] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [revision] => int
 [keepold] => int
 [structure] => string
 [timemodified] => string
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="imscps">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="revision">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="keepold">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="structure">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

`mod_imscp_view_imscp`

Simulate the view.php web interface imscp: trigger events, completion, etc...



### Arguments

`imscpid` (Required)

imscp instance id

#### General structure

```
int //imscp instance id
```

#### XML-RPC (PHP structure)

```
[imscpid] => int
```

#### REST (POST parameters)

```
imscpid= int
```

**Response**

**General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 /warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_label\_get\_labels\_by\_courses**

Returns a list of labels in a provided list of courses, if no list is provided all labels that the user can view will be returned.

**Arguments****courseids** (Default to "Array ()")

Array of course ids

**General structure**

```
Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)
```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [0] => int
)
```

**REST (POST parameters)**

courseids[0]= int

**Response**

**General structure**

```
object {
labels list of (
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
timemodified int //Last time the label was modified
}
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```



**XML-RPC (PHP structure)**

```
Array
(
 [labels] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [timemodified] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="labels">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

<KEY name="itemid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="warningcode">
 <VALUE>string</VALUE>
</KEY>
<KEY name="message">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### `mod_lesson_finish_attempt`

Finishes the current attempt.

### Arguments

#### `lessonid` (Required)

Lesson instance id.

**General structure**

```
int //Lesson instance id.
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

#### `password` (Default to "")

Optional password (the lesson may be protected).

**General structure**

```
string Default to "" //Optional password (the Lesson may be protected).
```

**XML-RPC (PHP structure)**

```
[password] => string
```

**REST (POST parameters)**

```
password= string
```

**outoftime** (Default to "")

If the user run out of time.

**General structure**

```
int Default to "" //If the user run out of time.
```

**XML-RPC (PHP structure)**

```
[outoftime] => int
```

**REST (POST parameters)**

```
outoftime= int
```

**review** (Default to "")

If we want to review just after finishing (1 hour margin).

**General structure**

```
int Default to "" //If we want to review just after finishing (1 hour margin).
```

**XML-RPC (PHP structure)**

```
[review] => int
```

**REST (POST parameters)**

```
review= int
```

**Response**



**General structure**

```

object {
 data //The EOL page information data.
 list of (
 object {
 name string //Data name.
 value string //Data value.
 message string //Data message (translated string).
 }
)messages list of (
 //The Lesson generated messages
 object {
 message string //Message.
 type string //Message type: usually a CSS identifier like:
 success, info, warning, error, notifyproblem, notifyerror, notifytiny, notifysuccess
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [data] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
 [message] => string
)
)
 [messages] =>
 Array
 (
 [0] =>
 Array
 (
 [message] => string
 [type] => string
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="data">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="messages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

### mod\_lesson\_get\_attempts\_overview

Get a list of all the attempts made by users in a lesson.

#### Arguments

**lessonid** (Required)

lesson instance id

##### General structure

```
int //Lesson instance id
```

##### XML-RPC (PHP structure)

```
[lessonid] => int
```

##### REST (POST parameters)

```
lessonid= int
```

**groupid** (Default to "0")

group id, 0 means that the function will determine the user group

##### General structure

```
int Default to "0" //group id, 0 means that the function will determine the user group
```

##### XML-RPC (PHP structure)

```
[groupid] => int
```

##### REST (POST parameters)

```
groupid= int
```

#### Response



**General structure**

```
object {
data Optional //Attempts overview data (empty for no attempts).
object {
lessonscored int //True if the lesson was scored.
numofattempts int //Number of attempts.
avescore double //Average score.
highscore double //High score.
lowscore double //Low score.
avetime int //Average time (spent in taking the lesson).
hightime int //High time.
lowtime int //Low time.
students Optional //Students data, including attempts.
list of (
object {
id int //User id.
fullname string //User full name.
bestgrade double //Best grade.
attempts list of (
object {
try int //Attempt number.
grade double //Attempt grade.
timestart int //Attempt time started.
timeend int //Attempt last time continued.
end int //Attempt time ended.
})
})
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```



## XML-RPC (PHP structure)

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="data">
 <SINGLE>
 <KEY name="lessonscored">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="numoffattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="avescore">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="highscore">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="lowscore">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="avetime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hightime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lowtime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="students">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="bestgrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="try">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="timestart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeend">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="end">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_content\_pages\_viewed**

Return the list of content pages viewed by a user during a lesson attempt.

**Arguments****lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**lessonattempt** (Required)

lesson attempt number

**General structure**

```
int //Lesson attempt number
```

**XML-RPC (PHP structure)**

```
[lessonattempt] => int
```

**REST (POST parameters)**

lessonattempt= int

userid (Default to "null")

the user id (empty for current user)

**General structure**int **Default to "null"** //the user id (empty for current user)**XML-RPC (PHP structure)**

[userid] =&gt; int

**REST (POST parameters)**

userid= int

**Response****General structure**

```

object {
 pages list of (
 //The content pages viewed.
 object {
 id int //The attempt id.
 lessonid int //The Lesson id.
 pageid int //The page id.
 userid int //The user who viewed the page.
 retry int //The Lesson attempt number.
 flag int //1 if the next page was calculated randomly.
 timeseen int //The time the page was seen.
 nextpageid int //The next page chosen id.
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```
Array
(
 [pages] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [lessonid] => int
 [pageid] => int
 [userid] => int
 [retry] => int
 [flag] => int
 [timeseen] => int
 [nextpageid] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="retry">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeseen">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpageid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

### mod\_lesson\_get\_lesson

Return information of a given lesson.

#### Arguments

lessonid (Required)

lesson instance id

##### General structure

```
int //Lesson instance id
```

##### XML-RPC (PHP structure)

```
[lessonid] => int
```

##### REST (POST parameters)

```
lessonid= int
```

password (Default to "")

lesson password

##### General structure

```
string Default to "" //Lesson password
```

##### XML-RPC (PHP structure)

```
[password] => string
```

##### REST (POST parameters)

```
password= string
```

#### Response



**General structure**

```

object {
lesson object {
id int //Standard Moodle primary key.
course int //Foreign key reference to the course this Lesson is part of.
coursemodule int //Course module id.
name string //Lesson name.
intro string Optional //Lesson introduction text.
introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
lang string //Forced activity Language
practice int Optional //Practice Lesson?
modattempts int Optional //Allow student review?
usepassword int Optional //Password protected Lesson?
password string Optional //Password
dependency int Optional //Dependent on (another Lesson id)
conditions string Optional //Conditions to enable the Lesson
grade int Optional //The total that the grade is scaled to be out of
custom int Optional //Custom scoring?
ongoing int Optional //Display ongoing score?
usemaxgrade int Optional //How to calculate the final grade
maxanswers int Optional //Maximum answers per page
maxattempts int Optional //Maximum attempts
review int Optional //Provide option to try a question again
nextpagedefault int Optional //Action for a correct answer
feedback int Optional //Display default feedback
minquestions int Optional //Minimum number of questions
maxpages int Optional //Number of pages to show
timelimit int Optional //Time Limit
retake int Optional //Re-takes allowed
activitylink int Optional //Id of the next activity to be linked once the Lesson is completed
mediafile string Optional //Local file path or full external URL
mediaheight int Optional //Popup for media file height
mediawidth int Optional //Popup for media width
mediaclose int Optional //Display a close button in the popup?
slideshow int Optional //Display lesson as slideshow
width int Optional //Slideshow width
height int Optional //Slideshow height
bgcolor string Optional //Slideshow bgcolor
displayleft int Optional //Display left pages menu?
displayleftif int Optional //Minimum grade to display menu
progressbar int Optional //Display progress bar?
available int Optional //Available from
deadline int Optional //Available until
timemodified int Optional //Last time settings were updated
completionendreached int Optional //Require end reached for completion?
completionspent int Optional //Student must do this activity at Least for
allowofflineattempts int //Whether to allow the Lesson to be attempted offline in the mobile app
introfiles Optional //introfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
)mediafiles Optional //mediafiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
))
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
}

```





^

**XML-RPC (PHP structure)**

```

Array
(
 [lesson] =>
 Array
 (
 [id] => int
 [course] => int
 [coursemodule] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [lang] => string
 [practice] => int
 [modattempts] => int
 [usepassword] => int
 [password] => string
 [dependency] => int
 [conditions] => string
 [grade] => int
 [custom] => int
 [ongoing] => int
 [usemaxgrade] => int
 [maxanswers] => int
 [maxattempts] => int
 [review] => int
 [nextpagedefault] => int
 [feedback] => int
 [minquestions] => int
 [maxpages] => int
 [timelimit] => int
 [retake] => int
 [activitylink] => int
 [mediafile] => string
 [mediaheight] => int
 [mediawidth] => int
 [mediaclose] => int
 [slideshow] => int
 [width] => int
 [height] => int
 [bgcolor] => string
 [displayleft] => int
 [displayleftif] => int
 [progressbar] => int
 [available] => int
 [deadline] => int
 [timemodified] => int
 [completionendreached] => int
 [completiontimespent] => int
 [allowofflineattempts] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [mediafiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)

```



```
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="lesson">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="practice">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="modattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usepassword">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="password">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="dependency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="conditions">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="custom">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ongoing">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usemaxgrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxanswers">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="review">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpagedefault">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedback">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="minquestions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxpages">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timelimit">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="retake">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="activitylink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediafile">
 <VALUE>string</VALUE>
</KEY>
<KEY name="mediaheight">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediawidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediaclose">
 <VALUE>int</VALUE>
</KEY>
<KEY name="slideshow">
 <VALUE>int</VALUE>
</KEY>
<KEY name="width">
 <VALUE>int</VALUE>
</KEY>
<KEY name="height">
 <VALUE>int</VALUE>
</KEY>
<KEY name="bgcolor">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displayleft">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displayleftif">
 <VALUE>int</VALUE>
</KEY>
<KEY name="progressbar">
 <VALUE>int</VALUE>
</KEY>
<KEY name="available">
 <VALUE>int</VALUE>
</KEY>
<KEY name="deadline">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completionendreached">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completiontimespent">
 <VALUE>int</VALUE>
</KEY>
<KEY name="allowofflineattempts">
 <VALUE>int</VALUE>
</KEY>
<KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="mediafiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_lesson\_get\_lesson\_access\_information

Return access information for a given lesson.

#### Arguments

lessonid (Required)

lesson instance id

#### General structure

```
int //Lesson instance id
```

#### XML-RPC (PHP structure)

```
[lessonid] => int
```

#### REST (POST parameters)

```
lessonid= int
```

#### Response

#### General structure

```
object {
 canmanage int //Whether the user can manage the Lesson or not.
 cangrade int //Whether the user can grade the Lesson or not.
 canviewreports int //Whether the user can view the Lesson reports or not.
 reviewmode int //Whether the Lesson is in review mode for the current user.
 attemptscount int //The number of attempts done by the user.
 lastpageseen int //The last page seen id.
 leftduringtimedsession int //Whether the user left during a timed session.
 firstpageid int //The Lesson first page id.
 preventaccessreasons list of (
 //The reasons why the user cannot attempt the Lesson
 object {
 reason string //Reason lang string code
 data string //Additional data
 message string //Complete html message
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [canmanage] => int
 [cangrade] => int
 [canviewreports] => int
 [reviewmode] => int
 [attemptscount] => int
 [lastpageseen] => int
 [leftduringtimedsession] => int
 [firstpageid] => int
 [preventaccessreasons] =>
 Array
 (
 [0] =>
 Array
 (
 [reason] => string
 [data] => string
 [message] => string
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cangrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewreports">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attemptscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lastpageseen">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="leftduringtimedsession">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="firstpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preventaccessreasons">
 <MULTIPLE>
 <SINGLE>
 <KEY name="reason">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="data">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

## REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

### mod\_lesson\_get\_lessons\_by\_courses

Returns a list of lessons in a provided list of courses, if no list is provided all lessons that the user can view will be returned.

#### Arguments

courseids (Default to "Array ()")

Array of course ids

##### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

##### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

##### REST (POST parameters)

```
courseids[0]= int
```

#### Response



**General structure**

```

object {
lessons list of (
object {
id int //Standard Moodle primary key.
course int //Foreign key reference to the course this lesson is part of.
coursemodule int //Course module id.
name string //Lesson name.
intro string Optional //Lesson introduction text.
introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
lang string //Forced activity language
practice int Optional //Practice lesson?
modattempts int Optional //Allow student review?
usepassword int Optional //Password protected lesson?
password string Optional //Password
dependency int Optional //Dependent on (another lesson id)
conditions string Optional //Conditions to enable the lesson
grade int Optional //The total that the grade is scaled to be out of
custom int Optional //Custom scoring?
ongoing int Optional //Display ongoing score?
usemaxgrade int Optional //How to calculate the final grade
maxanswers int Optional //Maximum answers per page
maxattempts int Optional //Maximum attempts
review int Optional //Provide option to try a question again
nextpagedefault int Optional //Action for a correct answer
feedback int Optional //Display default feedback
minquestions int Optional //Minimum number of questions
maxpages int Optional //Number of pages to show
timelimit int Optional //Time limit
retake int Optional //Re-takes allowed
activitylink int Optional //Id of the next activity to be linked once the lesson is completed
mediafile string Optional //Local file path or full external URL
mediaheight int Optional //Popup for media file height
mediawidth int Optional //Popup for media width
mediaclose int Optional //Display a close button in the popup?
slideshow int Optional //Display lesson as slideshow
width int Optional //Slideshow width
height int Optional //Slideshow height
bgcolor string Optional //Slideshow bgcolor
displayleft int Optional //Display left pages menu?
displayleftif int Optional //Minimum grade to display menu
progressbar int Optional //Display progress bar?
available int Optional //Available from
deadline int Optional //Available until
timemodified int Optional //Last time settings were updated
completionendreached int Optional //Require end reached for completion?
completiontimespent int Optional //Student must do this activity at least for
allowofflineattempts int //Whether to allow the lesson to be attempted offline in the mobile app
introfiles Optional //introfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
)mediafiles Optional //mediafiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
))
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
}

```





^

**XML-RPC (PHP structure)**

```

Array
(
 [lessons] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [course] => int
 [coursemodule] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [lang] => string
 [practice] => int
 [modattempts] => int
 [usepassword] => int
 [password] => string
 [dependency] => int
 [conditions] => string
 [grade] => int
 [custom] => int
 [ongoing] => int
 [usemaxgrade] => int
 [maxanswers] => int
 [maxattempts] => int
 [review] => int
 [nextpagedefault] => int
 [feedback] => int
 [minquestions] => int
 [maxpages] => int
 [timelimit] => int
 [retake] => int
 [activitylink] => int
 [mediafile] => string
 [mediaheight] => int
 [mediawidth] => int
 [mediaclose] => int
 [slideshow] => int
 [width] => int
 [height] => int
 [bgcolor] => string
 [displayleft] => int
 [displayleftif] => int
 [progressbar] => int
 [available] => int
 [deadline] => int
 [timemodified] => int
 [completionendreached] => int
 [completionspent] => int
 [allowofflineattempts] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
)
)
[mediafiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
)
)
)

```

```
[isexternalfile] => int
[repositorytype] => string
)
)
)
)
)
[warnings] =>
Array
(
[
[0] =>
Array
(
[
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="lessons">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="practice">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="modattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usepassword">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="password">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="dependency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="conditions">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="custom">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ongoing">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usemaxgrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxanswers">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="review">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpagedefault">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedback">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="minquestions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxpages">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timelimit">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="retake">
 <VALUE>int</VALUE>
</KEY>
<KEY name="activitylink">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediafile">
 <VALUE>string</VALUE>
</KEY>
<KEY name="mediaheight">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediawidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mediaclose">
 <VALUE>int</VALUE>
</KEY>
<KEY name="slideshow">
 <VALUE>int</VALUE>
</KEY>
<KEY name="width">
 <VALUE>int</VALUE>
</KEY>
<KEY name="height">
 <VALUE>int</VALUE>
</KEY>
<KEY name="bgcolor">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displayleft">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displayleftif">
 <VALUE>int</VALUE>
</KEY>
<KEY name="progressbar">
 <VALUE>int</VALUE>
</KEY>
<KEY name="available">
 <VALUE>int</VALUE>
</KEY>
<KEY name="deadline">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completionendreached">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completiontimespent">
 <VALUE>int</VALUE>
</KEY>
<KEY name="allowofflineattempts">
 <VALUE>int</VALUE>
</KEY>
<KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
```

^

```

 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="mediafiles">
<MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

**mod\_lesson\_get\_page\_data**

Return information of a given page, including its contents.

**Arguments****lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**pageid** (Required)

the page id

**General structure**

```
int //the page id
```

**XML-RPC (PHP structure)**

```
[pageid] => int
```

**REST (POST parameters)**

```
pageid= int
```

**password** (Default to "")

optional password (the lesson may be protected)

**General structure**

```
string Default to "" //optional password (the Lesson may be protected)
```

**XML-RPC (PHP structure)**

```
[password] => string
```

**REST (POST parameters)**

```
password= string
```

**review** (Default to "")

if we want to review just after finishing (1 hour margin)

**General structure**

```
int Default to "" //if we want to review just after finishing (1 hour margin)
```

**XML-RPC (PHP structure)**

```
[review] => int
```

**REST (POST parameters)**

```
review= int
```

**returncontents (Default to "")**

if we must return the complete page contents once rendered

**General structure**

```
int Default to "" //if we must return the complete page contents once rendered
```

**XML-RPC (PHP structure)**

```
[returncontents] => int
```

**REST (POST parameters)**

```
returncontents= int
```

**Response**

**General structure**

```

object {
page Optional //Page fields
object {
id int //The id of this Lesson page
lessonid int //The id of the Lesson this page belongs to
prevpageid int //The id of the page before this one
nextpageid int //The id of the next page in the page sequence
qtype int //Identifies the page type of this page
qoption int //Used to record page type specific options
layout int //Used to record page specific layout selections
display int //Used to record page specific display selections
timecreated int //Timestamp for when the page was created
timemodified int //Timestamp for when the page was last modified
title string Optional //The title of this page
contents string Optional //The contents of this page
contentsformat int Optional //contents format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
displayinmenublock int //Toggles display in the left menu block
type int //The type of the page [question | structure]
typeid int //The unique identifier for the page type
typestring string //The string that describes this page type
}
newpageid int //New page id (if a jump was made)
pagecontent string Optional //Page html content
ongoingscore string //The ongoing score message
progress int //Progress percentage in the Lesson
contentfiles //List of files.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)answers list of (
//The page answers
object {
id int //The ID of this answer in the database
answerfiles //List of files.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)responsefiles //List of files.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
)jumpto int Optional //Identifies where the user goes upon completing a page with this answer
grade int Optional //The grade this answer is worth
score int Optional //The score this answer will give
flags int Optional //Used to store options for the answer
timecreated int Optional //A timestamp of when the answer was created
timemodified int Optional //A timestamp of when the answer was modified
answer string Optional //Possible answer text
answerformat int Optional //answer format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
response string Optional //Response text for the answer
responseformat int Optional //response format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
)
)messages list of (
//The Lesson generated messages

```

```
object {
 message string //Message.
 type string //Message type: usually a CSS identifier Like:
 success, info, warning, error, notifyproblem, notifyerror, notifytiny, notifysuccess
}
)displaymenu int //Whether we should display the menu or not in this page.
warnings Optional //list of warnings
list of (
 //warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [page] =>
 Array
 (
 [id] => int
 [lessonid] => int
 [prevpageid] => int
 [nextpageid] => int
 [qtype] => int
 [option] => int
 [layout] => int
 [display] => int
 [timecreated] => int
 [timemodified] => int
 [title] => string
 [contents] => string
 [contentsformat] => int
 [displayinmenublock] => int
 [type] => int
 [typeid] => int
 [typestring] => string
)
 [newpageid] => int
 [pagecontent] => string
 [ongoingscore] => string
 [progress] => int
 [contentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [answers] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [answerfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [responsefiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)

```



```
)
)
[jumpto] => int
[grade] => int
[score] => int
[flags] => int
[timecreated] => int
[timemodified] => int
[answer] => string
[answerformat] => int
[response] => string
[responseformat] => int
)
)
)
[messages] =>
Array
(
 [
 [0] =>
 Array
 (
 [
 [message] => string
 [type] => string
)
)
)
)
[displaymenu] => int
[warnings] =>
Array
(
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="page">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="prevpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="qtype">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ooption">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="display">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contents">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentsformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayinmenublock">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typeid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typestring">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="newpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pagecontent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ongoingscore">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </RESPONSE>
```



```
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="answers">
<MULTIPLE>
<SINGLE>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="answerfiles">
<MULTIPLE>
<SINGLE>
<KEY name="filename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="responsefiles">
<MULTIPLE>
<SINGLE>
<KEY name="filename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filepath">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
```



```
</MULTIPLE>
</KEY>
<KEY name="jumpTo">
<VALUE>int</VALUE>
</KEY>
<KEY name="grade">
<VALUE>int</VALUE>
</KEY>
<KEY name="score">
<VALUE>int</VALUE>
</KEY>
<KEY name="flags">
<VALUE>int</VALUE>
</KEY>
<KEY name="timeCreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timeModified">
<VALUE>int</VALUE>
</KEY>
<KEY name="answer">
<VALUE>string</VALUE>
</KEY>
<KEY name="answerFormat">
<VALUE>int</VALUE>
</KEY>
<KEY name="response">
<VALUE>string</VALUE>
</KEY>
<KEY name="responseFormat">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="messages">
<MULTIPLE>
<SINGLE>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
<KEY name="type">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="displayMenu">
<VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
<SINGLE>
<KEY name="item">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemId">
<VALUE>int</VALUE>
</KEY>
<KEY name="warningCode">
<VALUE>string</VALUE>
</KEY>
<KEY name="message">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

**mod\_lesson\_get\_pages**

Return the list of pages in a lesson (based on the user permissions).

**Arguments**

**lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**password** (Default to "")

optional password (the lesson may be protected)

**General structure**

```
string Default to "" //optional password (the Lesson may be protected)
```

**XML-RPC (PHP structure)**

```
[password] => string
```

**REST (POST parameters)**

```
password= string
```

**Response**

**General structure**

```
object {
 pages list of (
 //The Lesson pages
 object {
 page //Page fields
 object {
 id int //The id of this Lesson page
 lessonid int //The id of the Lesson this page belongs to
 prevpageid int //The id of the page before this one
 nextpageid int //The id of the next page in the page sequence
 qtype int //Identifies the page type of this page
 qoption int //Used to record page type specific options
 layout int //Used to record page specific layout selections
 display int //Used to record page specific display selections
 timcreated int //Timestamp for when the page was created
 timemodified int //Timestamp for when the page was last modified
 title string Optional //The title of this page
 contents string Optional //The contents of this page
 contentsformat int Optional //contents format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 displayinmenublock int //Toggles display in the left menu block
 type int //The type of the page [question | structure]
 typeid int //The unique identifier for the page type
 typestring string //The string that describes this page type
 }
 answerids //List of answers ids (empty for content pages in Moodle 1.9)
 list of (
 int //Answer id
)jumps //List of possible page jumps
 list of (
 int //Page to jump id
)filescount int //The total number of files attached to the page
 filessizetotal int //The total size of the files
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
))
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [pages] =>
 Array
 (
 [0] =>
 Array
 (
 [page] =>
 Array
 (
 [id] => int
 [lessonid] => int
 [prevpageid] => int
 [nextpageid] => int
 [qtype] => int
 [qoption] => int
 [layout] => int
 [display] => int
 [timecreated] => int
 [timemodified] => int
 [title] => string
 [contents] => string
 [contentsformat] => int
 [displayinmenublock] => int
 [type] => int
 [typeid] => int
 [typestring] => string
)
 [answerids] =>
 Array
 (
 [0] => int
)
 [jumps] =>
 Array
 (
 [0] => int
)
 [filecount] => int
 [filessizetotal] => int
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="page">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="prevpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="qtype">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="qoption">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="display">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contents">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentsformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayinmenublock">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typeid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typestring">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="answerids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="jumps">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="filescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filessizetotal">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
```



```

<MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

**REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_lesson\_get\_pages\_possible\_jumps

Return all the possible jumps for the pages in a given lesson.

### Arguments

**lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

### Response

**General structure**

```
object {
jumps list of (
 //Jump for a page answer
object {
pageid int //The page id
answerid int //The answer id
jumpto int //The jump (page id or type of jump)
calculatedjump int //The real page id (or EOL) to jump
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
})}
```

**XML-RPC (PHP structure)**

```
Array
(
[jumps] =>
 Array
 (
 [0] =>
 Array
 (
 [pageid] => int
 [answerid] => int
 [jumpto] => int
 [calculatedjump] => int
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="jumps">
 <MULTIPLE>
 <SINGLE>
 <KEY name="pageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="answerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="jumpsto">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="calculatedjump">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_questions\_attempts**

Return the list of questions attempts in a given lesson.

**Arguments**

lessonid (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**attempt** (Required)

lesson attempt number

**General structure**

```
int //Lesson attempt number
```

**XML-RPC (PHP structure)**

```
[attempt] => int
```

**REST (POST parameters)**

```
attempt= int
```

**correct** (Default to "")

only fetch correct attempts

**General structure**

```
int Default to "" //only fetch correct attempts
```

**XML-RPC (PHP structure)**

```
[correct] => int
```

**REST (POST parameters)**

```
correct= int
```



**pageid** (Default to "null")

only fetch attempts at the given page

**General structure**

```
int Default to "null" //only fetch attempts at the given page
```

**XML-RPC (PHP structure)**

```
[pageid] => int
```

**REST (POST parameters)**

```
pageid= int
```



**userid** (Default to "null")

only fetch attempts of the given user

**General structure**

```
int Default to "null" //only fetch attempts of the given user
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response****General structure**

```
object {
 attempts list of (
 //The question page attempts
 object {
 id int //The attempt id
 lessonid int //The attempt Lessonid
 pageid int //The attempt pageid
 userid int //The user who did the attempt
 answerid int //The attempt answerid
 retry int //The lesson attempt number
 correct int //If it was the correct answer
 useranswer string //The complete user answer
 timeseen int //The time the question was seen
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [attempts] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [lessonid] => int
 [pageid] => int
 [userid] => int
 [answerid] => int
 [retry] => int
 [correct] => int
 [useranswer] => string
 [timeseen] => int
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="answerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="retry">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="correct">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="useranswer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timeseen">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

## REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_user\_attempt**

Return information about the given user attempt (including answers).

**Arguments****lessonid** (Required)

Lesson instance id.

**General structure**

```
int //Lesson instance id.
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**userid** (Required)

The user id. 0 for current user.

**General structure**

```
int //The user id. 0 for current user.
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**lessonattempt** (Required)

The attempt number.

**General structure**

```
int //The attempt number.
```

**XML-RPC (PHP structure)**

```
[lessonattempt] => int
```

**REST (POST parameters)**

```
lessonattempt= int
```

**Response**

**General structure**

```

object {
answerpages list of (
object {
page Optional //Page fields
object {
id int //The id of this Lesson page
lessonid int //The id of the Lesson this page belongs to
prevpageid int //The id of the page before this one
nextpageid int //The id of the next page in the page sequence
qtype int //Identifies the page type of this page
qoption int //Used to record page type specific options
layout int //Used to record page specific layout selections
display int //Used to record page specific display selections
timemigrated int //Timestamp for when the page was created
timemodified int //Timestamp for when the page was last modified
title string Optional //The title of this page
contents string Optional //The contents of this page
contentsformat int Optional //contents format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
displayinmenublock int //Toggles display in the left menu block
type int //The type of the page [question / structure]
typeid int //The unique identifier for the page type
typestring string //The string that describes this page type
}
title string //Page title.
contents string //Page contents.
qtype string //Identifies the page type of this page.
grayout int //If is required to apply a grayout.
answerdata Optional //Answer data (empty in content pages created in Moodle 1.x).
object {
score string //The score (text version).
response string //The response text.
responseformat int //response. format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
answers Optional //User answers
list of (
list of (
string //Possible answers and info.
)))
}
)
)userstats object {
grade double //Attempt final grade.
completed int //Time completed.
timetotake int //time taken.
gradeinfo Optional //Attempt grade
object {
nquestions int //Number of questions answered
attempts int //Number of question attempts
total double //Max points possible
earned double //Points earned by student
grade double //Calculated percentage grade
nmanual int //Number of manually graded questions
manualpoints double //Point value for manually graded questions
}
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [answerpages] =>
 Array
 (
 [0] =>
 Array
 (
 [page] =>
 Array
 (
 [id] => int
 [lessonid] => int
 [prevpageid] => int
 [nextpageid] => int
 [qtype] => int
 [qoption] => int
 [layout] => int
 [display] => int
 [timecreated] => int
 [timemodified] => int
 [title] => string
 [contents] => string
 [contentsformat] => int
 [displayinmenublock] => int
 [type] => int
 [typeid] => int
 [typestring] => string
)
 [title] => string
 [contents] => string
 [qtype] => string
 [grayout] => int
 [answerdata] =>
 Array
 (
 [score] => string
 [response] => string
 [responseformat] => int
 [answers] =>
 Array
 (
 [0] =>
 Array
 (
 [0] => string
)
)
)
)
)
)
)
)
[userstats] =>
 Array
 (
 [grade] => double
 [completed] => int
 [timetotake] => int
 [gradeinfo] =>
 Array
 (
 [nquestions] => int
 [attempts] => int
 [total] => double
 [earned] => double
 [grade] => double
 [nmanual] => int
 [manualpoints] => double
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```





^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="answerpages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="page">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="prevpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nextpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="qtype">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="qoption">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="display">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contents">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentsformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayinmenublock">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typeid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typestring">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contents">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="qtype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="grayout">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="answerdata">
 <SINGLE>
 <KEY name="score">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="response">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="responseformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="answers">
 <MULTIPLE>
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </MULTIPLE>
</KEY>
<KEY name="userstats">
 <SINGLE>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="completed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timetotake">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradeinfo">
 <SINGLE>
 <KEY name="nquestions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="total">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="earned">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="nmanual">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="manualpoints">
 <VALUE>double</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>
```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_user\_attempt\_grade**

Return grade information in the attempt for a given user.

**Arguments****lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**lessonattempt** (Required)

lesson attempt number

**General structure**

```
int //Lesson attempt number
```

**XML-RPC (PHP structure)**

```
[lessonattempt] => int
```

**REST (POST parameters)**

```
lessonattempt= int
```

**userid** (Default to "null")

the user id (empty for current user)

**General structure**

```
int Default to "null" //the user id (empty for current user)
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response**

**General structure**

```
object {
 grade //Attempt grade
 object {
 nquestions int //Number of questions answered
 attempts int //Number of question attempts
 total double //Max points possible
 earned double //Points earned by student
 grade double //Calculated percentage grade
 nmanual int //Number of manually graded questions
 manualpoints double //Point value for manually graded questions
 }
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [grade] =>
 Array
 (
 [nquestions] => int
 [attempts] => int
 [total] => double
 [earned] => double
 [grade] => double
 [nmanual] => int
 [manualpoints] => double
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <KEY name="grade">
 <KEY name="nquestions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="total">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="earned">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="nmanual">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="manualpoints">
 <VALUE>double</VALUE>
 </KEY>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <KEY name="item">
 <KEY name="itemid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </KEY>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_user\_grade**

Return the final grade in the lesson for the given user.

### Arguments

**lessonid** (Required)

lesson instance id

#### General structure

```
int //Lesson instance id
```

#### XML-RPC (PHP structure)

```
[lessonid] => int
```

#### REST (POST parameters)

```
lessonid= int
```

**userid** (Default to "null")

the user id (empty for current user)

#### General structure

```
int Default to "null" //the user id (empty for current user)
```

#### XML-RPC (PHP structure)

```
[userid] => int
```

#### REST (POST parameters)

```
userid= int
```

### Response

#### General structure

```
object {
 grade double //The Lesson final raw grade
 formattedgrade string //The Lesson final grade formatted
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```

Array
(
 [grade] => double
 [formattedgrade] => string
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="formattedgrade">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_get\_user\_timers**

Return the timers in the current lesson for the given user.

**Arguments**

**lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**userid** (Default to "null")

the user id (empty for current user)

**General structure**

```
int Default to "null" //the user id (empty for current user)
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response****General structure**

```
object {
 timers list of (
 //The timers
 object {
 id int //The attempt id
 lessonid int //The Lesson id
 userid int //The user id
 starttime int //First access time for a new timer session
 lessontime int //Last access time to the Lesson during the timer session
 completed int //If the Lesson for this timer was completed
 timemodifiedoffline int //Last modified time via webservices.
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [
 [timers] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [lessonid] => int
 [userid] => int
 [starttime] => int
 [lessontime] => int
 [completed] => int
 [timemodifiedoffline] => int
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="timers">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessonid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="starttime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lessontime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodifiedoffline">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

### mod\_lesson\_launch\_attempt

Starts a new attempt or continues an existing one.

#### Arguments

**lessonid** (Required)

lesson instance id

##### General structure

```
int //Lesson instance id
```

##### XML-RPC (PHP structure)

```
[lessonid] => int
```

##### REST (POST parameters)

```
lessonid= int
```

**password** (Default to "")

optional password (the lesson may be protected)

##### General structure

```
string Default to "" //optional password (the Lesson may be protected)
```

##### XML-RPC (PHP structure)

```
[password] => string
```

##### REST (POST parameters)

```
password= string
```

**pageid** (Default to "0")

page id to continue from (only when continuing an attempt)

##### General structure

```
int Default to "0" //page id to continue from (only when continuing an attempt)
```



##### XML-RPC (PHP structure)

```
[pageid] => int
```

##### REST (POST parameters)

```
pageid= int
```

**review** (Default to "")

if we want to review just after finishing

##### General structure

```
int Default to "" //if we want to review just after finishing
```

**XML-RPC (PHP structure)**

```
[review] => int
```

**REST (POST parameters)**

```
review= int
```

**Response****General structure**

```
object {
 messages list of (
 //The Lesson generated messages
 object {
 message string //Message.
 type string //Message type: usually a CSS identifier like:
 success, info, warning, error, notifyproblem, notifyerror, notifytiny, notifysuccess
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [messages] =>
 Array
 (
 [0] =>
 Array
 (
 [message] => string
 [type] => string
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="messages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_process\_page**

Processes page responses.

**Arguments****lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**pageid** (Required)

the page id

**General structure**

```
int //the page id
```

**XML-RPC (PHP structure)**

```
[pageid] => int
```

**REST (POST parameters)**

```
pageid= int
```

**data** (Required)

the data to be saved

**General structure**

```
//the data to be saved
list of (
object {
name string //data name
value string //data value
})
```

**XML-RPC (PHP structure)**

```
[data] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
```

**REST (POST parameters)**

```
data[0][name]= string
data[0][value]= string
```

**password** (Default to "")

optional password (the lesson may be protected)

**General structure**

```
string Default to "" //optional password (the Lesson may be protected)
```

**XML-RPC (PHP structure)**

```
[password] => string
```

**REST (POST parameters)**

```
password= string
```

**review** (Default to "")

if we want to review just after finishing (1 hour margin)

**General structure**

```
int Default to "" //if we want to review just after finishing (1 hour margin)
```

**XML-RPC (PHP structure)**

```
[review] => int
```

**REST (POST parameters)**

```
review= int
```

**Response****General structure**

```
object {
 newpageid int //New page id (if a jump was made).
 immediatejump int //Whether the page processing redirect directly to another page.
 nodefaultresponse int //Whether there is not a default response.
 feedback string //The response feedback.
 attemptsremaining int //Number of attempts remaining.
 correctanswer int //Whether the answer is correct.
 noanswer int //Whether there aren't answers.
 isessayquestion int //Whether it is an essay question.
 maxattemptsreached int //Whether we reached the maximum number of attempts.
 response string //The response.
 studentanswer string //The student answer.
 userresponse string //The user response.
 reviewmode int //Whether the user is reviewing.
 ongoingscore string //The ongoing score.
 progress int //Progress percentage in the lesson.
 displaymenu int //Whether we should display the menu or not in this page.
 messages list of (
 //The lesson generated messages
 object {
 message string //Message.
 type string //Message type: usually a CSS identifier like:
 success, info, warning, error, notifyproblem, notifyerror, notifytiny, notifysuccess
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated English message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [newpageid] => int
 [inmediatejump] => int
 [nodefaultresponse] => int
 [feedback] => string
 [attemptsremaining] => int
 [correctanswer] => int
 [noanswer] => int
 [isessayquestion] => int
 [maxattemptsreached] => int
 [response] => string
 [studentanswer] => string
 [userresponse] => string
 [reviewmode] => int
 [ongoingscore] => string
 [progress] => int
 [displaymenu] => int
 [messages] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [message] => string
 [type] => string
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="newpageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="immediatejump">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nodefaultresponse">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedback">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="attemptsremaining">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="correctanswer">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="noanswer">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isessayquestion">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maxattemptsreached">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="response">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="studentanswer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userresponse">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="reviewmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ongoingscore">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displaymenu">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="messages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </RESPONSE>
```



```
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION classe="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lesson\_view\_lesson**

Trigger the course module viewed event and update the module completion status.

**Arguments****lessonid** (Required)

lesson instance id

**General structure**

```
int //Lesson instance id
```

**XML-RPC (PHP structure)**

```
[lessonid] => int
```

**REST (POST parameters)**

```
lessonid= int
```

**password** (Default to "")

lesson password

**General structure**

```
string Default to "" //Lesson password
```

**XML-RPC (PHP structure)**

```
[password] => string
```

**REST (POST parameters)**

```
password= string
```

**Response**

**General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 /warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_lti_create_tool_proxy`

Create a tool proxy

**Arguments**

`name` (Default to "")

Tool proxy name

**General structure**

```
string Default to "" //Tool proxy name
```

**XML-RPC (PHP structure)**

```
[name] => string
```

**REST (POST parameters)**

```
name= string
```

`regurl` (Required)

Tool proxy registration URL

**General structure**

```
string //Tool proxy registration URL
```

**XML-RPC (PHP structure)**

```
[regurl] => string
```

**REST (POST parameters)**

```
regurl= string
```

`capabilityoffered` (Default to "Array ()")

Array of capabilities

```
General structure

Default to "Array
(
)
" //Array of capabilities
list of (
string //Tool proxy capabilities offered
)
```

**XML-RPC (PHP structure)**

```
[capabilityoffered] =>
Array
(
 [
 [0] => string
)
)
```

**REST (POST parameters)**

```
capabilityoffered[0]= string
```

serviceoffered (Default to "Array ()")

Array of services

```
General structure

Default to "Array
(
)
" //Array of services
list of (
string //Tool proxy services offered
)
```

**XML-RPC (PHP structure)**

```
[serviceoffered] =>
Array
(
 [
 [0] => string
)
)
```

**REST (POST parameters)**

```
serviceoffered[0]= string
```

**Response**

```
General structure

object {
id int //Tool proxy id
name string //Tool proxy name
regurl string //Tool proxy registration URL
state int //Tool proxy state
guid string //Tool proxy globally unique identifier
secret string //Tool proxy shared secret
vendorcode string //Tool proxy consumer code
capabilityoffered string //Tool proxy capabilities offered
serviceoffered string //Tool proxy services offered
toolproxy string //Tool proxy
timecreated int //Tool proxy time created
timemodified int //Tool proxy modified
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [id] => int
 [name] => string
 [regurl] => string
 [state] => int
 [guid] => string
 [secret] => string
 [vendorcode] => string
 [capabilityoffered] => string
 [serviceoffered] => string
 [toolproxy] => string
 [timecreated] => int
 [timemodified] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="regurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="guid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="secret">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="vendorcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="capabilityoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="serviceoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="toolproxy">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod\_lti\_create\_tool\_type

Create a tool type

#### Arguments

cartridgeurl (Default to "")

URL to cartridge to load tool information

**General structure**

```
string Default to "" //URL to cartridge to Load tool information
```

**XML-RPC (PHP structure)**

```
[cartridgeurl] => string
```

**REST (POST parameters)**

```
cartridgeurl= string
```

key (Default to "")

Consumer key

**General structure**

```
string Default to "" //Consumer key
```

**XML-RPC (PHP structure)**

```
[key] => string
```

**REST (POST parameters)**

```
key= string
```



secret (Default to "")

Shared secret

**General structure**

```
string Default to "" //Shared secret
```

**XML-RPC (PHP structure)**

```
[secret] => string
```

**REST (POST parameters)**

```
secret= string
```

## Response

Tool

```
General structure

//Tool
object {
id int //Tool type id
name string //Tool type name
description string //Tool type description
platformid string //Platform ID
clientid string //Client ID
deploymentid int //Deployment ID
urls object {
icon string //Tool type icon URL
edit string //Tool type edit URL
course string Optional //Tool type edit URL
publickeyset string //Public Keyset URL
accesstoken string //Access Token URL
authrequest string //Authorisation Request URL
}
state object {
text string //Tool type state name string
pending int //Is the state pending
configured int //Is the state configured
rejected int //Is the state rejected
unknown int //Is the state unknown
}
hascapabilitygroups int //Indicate if capabilitygroups is populated
capabilitygroups Default to "Array
(
)
" //Array of capability groups
list of (
string //Tool type capability groups enabled
)courseid int Default to "0" //Tool type course
instanceids Default to "Array
(
)
" //IDs for the LTI instances using this type
list of (
int //LTI instance ID
)instancecount int //The number of times this tool is being used
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [id] => int
 [name] => string
 [description] => string
 [platformid] => string
 [clientid] => string
 [deploymentid] => int
 [urls] =>
 Array
 (
 [icon] => string
 [edit] => string
 [course] => string
 [publickeyset] => string
 [accesstoken] => string
 [authrequest] => string
)
 [state] =>
 Array
 (
 [text] => string
 [pending] => int
 [configured] => int
 [rejected] => int
 [unknown] => int
)
 [hascapabilitygroups] => int
 [capabilitygroups] =>
 Array
 (
 [0] => string
)
 [courseid] => int
 [instanceids] =>
 Array
 (
 [0] => int
)
 [instancecount] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="platformid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="clientid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="deploymentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="urls">
 <SINGLE>
 <KEY name="icon">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="edit">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="publickeyset">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="accesstoken">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authrequest">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="state">
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pending">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="configured">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rejected">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unknown">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="hascapabilitygroups">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="capabilitygroups">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="instanceids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="instancecount">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_delete\_tool\_proxy****Delete a tool proxy****Arguments****id** (Required)

Tool proxy id

**General structure**

int //Tool proxy id

**XML-RPC (PHP structure)**

[id] =&gt; int

**REST (POST parameters)**

id= int

**Response****General structure**

```
object {
 id int //Tool proxy id
 name string //Tool proxy name
 regurl string //Tool proxy registration URL
 state int //Tool proxy state
 guid string //Tool proxy globally unique identifier
 secret string //Tool proxy shared secret
 vendorcode string //Tool proxy consumer code
 capabilityoffered string //Tool proxy capabilities offered
 serviceoffered string //Tool proxy services offered
 toolproxy string //Tool proxy
 timecreated int //Tool proxy time created
 timemodified int //Tool proxy modified
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [id] => int
 [name] => string
 [regurl] => string
 [state] => int
 [guid] => string
 [secret] => string
 [vendorcode] => string
 [capabilityoffered] => string
 [serviceoffered] => string
 [toolproxy] => string
 [timecreated] => int
 [timemodified] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="regurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="guid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="secret">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="vendorcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="capabilityoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="serviceoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="toolproxy">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

mod\_lti\_delete\_tool\_type

Delete a tool type

#### Arguments

id (Required)

Tool type id

General structure

```
int //Tool type id
```

XML-RPC (PHP structure)

```
[id] => int
```

REST (POST parameters)

```
id= int
```

#### Response

General structure

```
object {
id int //Tool type id
}
```

XML-RPC (PHP structure)

```
Array
(
 [id] => int
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_get\_ltis\_by\_courses**

Returns a list of external tool instances in a provided set of courses, if no courses are provided then all the external tool instances the user has access to will be returned.

**Arguments****courseids** (Default to "Array ()")

Array of course ids

**General structure**

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [0] => int
)
```

**REST (POST parameters)**

courseids[0]= int

**Response**

**General structure**

```

object {
ltis list of (
 //Tool
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string Optional //Activity introduction
introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
timecreated int Optional //Time of creation
timemodified int Optional //Time of last modification
typeid int Optional //Type id
toolurl string Optional //Tool url
securetoolurl string Optional //Secure tool url
instructorchoicesendname string Optional //Instructor choice send name
instructorchoicesendemailaddr int Optional //instructor choice send mail address
instructorchoiceallowroster int Optional //Instructor choice allow roster
instructorchoiceallowsetting int Optional //Instructor choice allow setting
instructorcustomparameters string Optional //instructor custom parameters
instructorchoiceacceptgrades int Optional //instructor choice accept grades
grade int Optional //Enable grades
launchcontainer int Optional //Launch container mode
resourcekey string Optional //Resource key
password string Optional //Shared secret
debuglaunch int Optional //Debug launch
showtitlelaunch int Optional //Show title launch
showdescriptionlaunch int Optional //Show description launch
servicesalt string Optional //Service salt
icon string Optional //Alternative icon URL
secureicon string Optional //Secure icon URL
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}

```



## XML-RPC (PHP structure)

```

Array
(
 [ltis] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [timecreated] => int
 [timemodified] => int
 [typeid] => int
 [toolurl] => string
 [securetoolurl] => string
 [instructorchoicesendname] => string
 [instructorchoicesendemailaddr] => int
 [instructorchoiceallowroster] => int
 [instructorchoiceallowsetting] => int
 [instructorcustomparameters] => string
 [instructorchoiceacceptgrades] => int
 [grade] => int
 [launchcontainer] => int
 [resourcekey] => string
 [password] => string
 [debuglaunch] => int
 [showtitlelaunch] => int
 [showdescriptionlaunch] => int
 [servicesalt] => string
 [icon] => string
 [secureicon] => string
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="ltis">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="typeid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="toolurl">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="securetoolurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="instructorchoicesendname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="instructorchoicesendemailaddr">
 <VALUE>int</VALUE>
</KEY>
<KEY name="instructorchoiceallowroster">
 <VALUE>int</VALUE>
</KEY>
<KEY name="instructorchoiceallowsetting">
 <VALUE>int</VALUE>
</KEY>
<KEY name="instructorcustomparameters">
 <VALUE>string</VALUE>
</KEY>
<KEY name="instructorchoiceacceptgrades">
 <VALUE>int</VALUE>
</KEY>
<KEY name="grade">
 <VALUE>int</VALUE>
</KEY>
<KEY name="launchcontainer">
 <VALUE>int</VALUE>
</KEY>
<KEY name="resourcekey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="password">
 <VALUE>string</VALUE>
</KEY>
<KEY name="debuglaunch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showtitlelaunch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showdescriptionlaunch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="servicesalt">
 <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
 <VALUE>string</VALUE>
</KEY>
<KEY name="secureicon">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lti\_get\_tool\_launch\_data**

Return the launch data for a given external tool.

**Arguments****toolid** (Required)

external tool instance id

**General structure**

```
int //external tool instance id
```

**XML-RPC (PHP structure)**

```
[toolid] => int
```

**REST (POST parameters)**

```
toolid= int
```

**Response****General structure**

```
object {
endpoint string //Endpoint URL
parameters list of (
object {
name string //Parameter name
value string //Parameter value
})
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [endpoint] => string
 [parameters] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [value] => string
)
)
]
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
]
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="endpoint">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="parameters">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_lti\_get\_tool\_proxies**

Get a list of the tool proxies

**Arguments****orphanedonly** (Default to "0")

Orphaned tool types only

**General structure**

```
int Default to "0" //Orphaned tool types only
```

**XML-RPC (PHP structure)**

```
[orphanedonly] => int
```

**REST (POST parameters)**

```
orphanedonly= int
```

**Response****General structure**

```
list of (
object {
id int //Tool proxy id
name string //Tool proxy name
regurl string //Tool proxy registration URL
state int //Tool proxy state
guid string //Tool proxy globally unique identifier
secret string //Tool proxy shared secret
vendorcode string //Tool proxy consumer code
capabilityoffered string //Tool proxy capabilities offered
serviceoffered string //Tool proxy services offered
toolproxy string //Tool proxy
timecreated int //Tool proxy time created
timemodified int //Tool proxy modified
}
)
```

## XML-RPC (PHP structure)

```
Array
(
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [regurl] => string
 [state] => int
 [guid] => string
 [secret] => string
 [vendorcode] => string
 [capabilityoffered] => string
 [serviceoffered] => string
 [toolproxy] => string
 [timecreated] => int
 [timemodified] => int
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="regurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="guid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="secret">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="vendorcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="capabilityoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="serviceoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="toolproxy">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</RESPONSE>
```

## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_get\_tool\_proxy\_registration\_request**

Get a registration request for a tool proxy

**Arguments**

id (Required)

Tool proxy id

**General structure**

```
int //Tool proxy id
```

**XML-RPC (PHP structure)**

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

**Response****General structure**

```
object {
 lti_message_type string //LTI message type
 lti_version string //LTI version
 reg_key string //Tool proxy registration key
 reg_password string //Tool proxy registration password
 reg_url string //Tool proxy registration url
 tc_profile_url string //Tool consumers profile URL
 launch_presentation_return_url string //URL to redirect on registration completion
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [lti_message_type] => string
 [lti_version] => string
 [reg_key] => string
 [reg_password] => string
 [reg_url] => string
 [tc_profile_url] => string
 [launch_presentation_return_url] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="lti_message_type">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="lti_version">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="reg_key">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="reg_password">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="reg_url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="tc_profile_url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="launch_presentation_return_url">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes



mod\_lti\_get\_tool\_types

Get a list of the tool types

**Arguments**

toolproxyid (Default to "0")

Tool proxy id

**General structure**int **Default to "0"** //Tool proxy id**XML-RPC (PHP structure)**

[toolproxyid] =&gt; int

## REST (POST parameters)

toolproxyid= int

## Response

```
General structure

list of (
 //Tool
object {
 id int //Tool type id
 name string //Tool type name
 description string //Tool type description
 platformid string //Platform ID
 clientid string //Client ID
 deploymentid int //Deployment ID
 urls object {
 icon string //Tool type icon URL
 edit string //Tool type edit URL
 course string Optional //Tool type edit URL
 publickeyset string //Public Keyset URL
 accesstoken string //Access Token URL
 authrequest string //Authorisation Request URL
 }
 state object {
 text string //Tool type state name string
 pending int //Is the state pending
 configured int //Is the state configured
 rejected int //Is the state rejected
 unknown int //Is the state unknown
 }
 hascapabilitygroups int //Indicate if capabilitygroups is populated
 capabilitygroups Default to "Array"
 (
)
 " //Array of capability groups
 list of (
 string //Tool type capability groups enabled
)courseid int Default to "0" //Tool type course
 instanceids Default to "Array"
 (
)
 " //IDs for the LTI instances using this type
 list of (
 int //LTI instance ID
)instancecount int //The number of times this tool is being used
)
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [description] => string
 [platformid] => string
 [clientid] => string
 [deploymentid] => int
 [urls] =>
 Array
 (
 [icon] => string
 [edit] => string
 [course] => string
 [publickeyset] => string
 [accesstoken] => string
 [authrequest] => string
)
 [state] =>
 Array
 (
 [text] => string
 [pending] => int
 [configured] => int
 [rejected] => int
 [unknown] => int
)
 [hascapabilitygroups] => int
 [capabilitygroups] =>
 Array
 (
 [0] => string
)
 [courseid] => int
 [instanceids] =>
 Array
 (
 [0] => int
)
 [instancecount] => int
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="platformid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="clientid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="deploymentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="urls">
 <SINGLE>
 <KEY name="icon">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="edit">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="publickeyset">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="accesstoken">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authrequest">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 <KEY name="state">
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pending">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="configured">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rejected">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unknown">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="hascapabilitygroups">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="capabilitygroups">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="instanceids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="instancecount">
 <VALUE>int</VALUE>
 </KEY>

```



```
</KEY>
</SINGLE>
</MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_get\_tool\_types\_and\_proxies**

Get a list of the tool types and tool proxies

**Arguments**

toolproxyid (Default to "0")

Tool proxy id

**General structure**

```
int Default to "0" //Tool proxy id
```

**XML-RPC (PHP structure)**

```
[toolproxyid] => int
```

**REST (POST parameters)**

```
toolproxyid= int
```

orphanedonly (Default to "0")

Orphaned tool types only

**General structure**

```
int Default to "0" //Orphaned tool types only
```

**XML-RPC (PHP structure)**

```
[orphanedonly] => int
```

**REST (POST parameters)**

```
orphanedonly= int
```

**limit** (Default to "60")

How many tool types displayed per page

**General structure**

```
int Default to "60" //How many tool types displayed per page
```

**XML-RPC (PHP structure)**

```
[limit] => int
```

**REST (POST parameters)**

```
limit= int
```

**offset** (Default to "0")

Current offset of tool elements

**General structure**

```
int Default to "0" //Current offset of tool elements
```

**XML-RPC (PHP structure)**

```
[offset] => int
```

**REST (POST parameters)**

```
offset= int
```

**Response**

**General structure**

```

object {
types list of (
 //Tool
object {
id int //Tool type id
name string //Tool type name
description string //Tool type description
platformid string //Platform ID
clientid string //Client ID
deploymentid int //Deployment ID
urls object {
icon string //Tool type icon URL
edit string //Tool type edit URL
course string Optional //Tool type edit URL
publickeyset string //Public Keyset URL
accesstoken string //Access Token URL
authrequest string //Authorisation Request URL
}
state object {
text string //Tool type state name string
pending int //Is the state pending
configured int //Is the state configured
rejected int //Is the state rejected
unknown int //Is the state unknown
}
hascapabilitygroups int //Indicate if capabilitygroups is populated
capabilitygroups Default to "Array
(
)
" //Array of capability groups
list of (
string //Tool type capability groups enabled
)courseid int Default to "0" //Tool type course
instanceids Default to "Array
(
)
" //IDs for the LTI instances using this type
list of (
int //LTI instance ID
)instancecount int //The number of times this tool is being used
)
proxies list of (
object {
id int //Tool proxy id
name string //Tool proxy name
regurl string //Tool proxy registration URL
state int //Tool proxy state
guid string //Tool proxy globally unique identifier
secret string //Tool proxy shared secret
vendorcode string //Tool proxy consumer code
capabilityoffered string //Tool proxy capabilities offered
serviceoffered string //Tool proxy services offered
toolproxy string //Tool proxy
timecreated int //Tool proxy time created
timemodified int //Tool proxy modified
}
)limit int Optional //Limit of how many tool types to show
offset int Optional //Offset of tool types
}

```



**XML-RPC (PHP structure)**

```

Array
(
[types] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [description] => string
 [platformid] => string
 [clientid] => string
 [deploymentid] => int
 [urls] =>
 Array
 (
 [icon] => string
 [edit] => string
 [course] => string
 [publickeyset] => string
 [accesstoken] => string
 [authrequest] => string
)
 [state] =>
 Array
 (
 [text] => string
 [pending] => int
 [configured] => int
 [rejected] => int
 [unknown] => int
)
 [hascapabilitygroups] => int
 [capabilitygroups] =>
 Array
 (
 [0] => string
)
 [courseid] => int
 [instanceids] =>
 Array
 (
 [0] => int
)
 [instancecount] => int
)
)
 [proxies] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [regurl] => string
 [state] => int
 [guid] => string
 [secret] => string
 [vendorcode] => string
 [capabilityoffered] => string
 [serviceoffered] => string
 [toolproxy] => string
 [timecreated] => int
 [timemodified] => int
)
)
 [limit] => int
 [offset] => int
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="types">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="platformid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="clientid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="deploymentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="urls">
 <SINGLE>
 <KEY name="icon">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="edit">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="publickeyset">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="accesstoken">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authrequest">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="state">
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pending">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="configured">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rejected">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unknown">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="hascapabilitygroups">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="capabilitygroups">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="instanceids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="instancecount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proxies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="regurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="guid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="secret">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="vendorcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="capabilityoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="serviceoffered">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="toolproxy">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="limit">
 <VALUE>int</VALUE>
</KEY>
<KEY name="offset">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

### mod\_lti\_get\_tool\_types\_and\_proxies\_count

Get total number of the tool types and tool proxies

#### Arguments

toolproxyid (Default to "0")

Tool proxy id

##### General structure

```
int Default to "0" //Tool proxy id
```

##### XML-RPC (PHP structure)

```
[toolproxyid] => int
```

##### REST (POST parameters)

```
toolproxyid= int
```

orphanedonly (Default to "0")

Orphaned tool types only

##### General structure

```
int Default to "0" //Orphaned tool types only
```

##### XML-RPC (PHP structure)

```
[orphanedonly] => int
```

##### REST (POST parameters)

```
orphanedonly= int
```

#### Response

##### General structure

```
object {
count int //Total number of tool types and proxies
}
```

##### XML-RPC (PHP structure)

```
Array
(
[count] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_is\_cartridge**

Determine if the given url is for a cartridge

**Arguments****url** (Required)

Tool url

**General structure**

```
string //Tool url
```

**XML-RPC (PHP structure)**

```
[url] => string
```

**REST (POST parameters)**

```
url= string
```

**Response****General structure**

```
object {
 iscartridge int //True if the URL is a cartridge
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [iscartridge] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="iscartridge">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_lti\_update\_tool\_type**

Update a tool type

**Arguments**

id (Required)

Tool type id

**General structure**

```
int //Tool type id
```

**XML-RPC (PHP structure)**

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

name (Default to "null")

Tool type name

**General structure**

```
string Default to "null" //Tool type name
```

**XML-RPC (PHP structure)**

```
[name] => string
```

**REST (POST parameters)**

```
name= string
```

**description** (Default to "null")

Tool type description

**General structure**

```
string Default to "null" //Tool type description
```

**XML-RPC (PHP structure)**

```
[description] => string
```

**REST (POST parameters)**

```
description= string
```

**state** (Default to "null")

Tool type state

**General structure**

```
int Default to "null" //Tool type state
```

**XML-RPC (PHP structure)**

```
[state] => int
```

**REST (POST parameters)**

```
state= int
```



## Response

Tool

```
General structure

//Tool
object {
id int //Tool type id
name string //Tool type name
description string //Tool type description
platformid string //Platform ID
clientid string //Client ID
deploymentid int //Deployment ID
urls object {
icon string //Tool type icon URL
edit string //Tool type edit URL
course string Optional //Tool type edit URL
publickeyset string //Public Keyset URL
accesstoken string //Access Token URL
authrequest string //Authorisation Request URL
}
state object {
text string //Tool type state name string
pending int //Is the state pending
configured int //Is the state configured
rejected int //Is the state rejected
unknown int //Is the state unknown
}
hascapabilitygroups int //Indicate if capabilitygroups is populated
capabilitygroups Default to "Array
(
)
" //Array of capability groups
list of (
string //Tool type capability groups enabled
)courseid int Default to "0" //Tool type course
instanceids Default to "Array
(
)
" //IDs for the LTI instances using this type
list of (
int //LTI instance ID
)instancecount int //The number of times this tool is being used
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [id] => int
 [name] => string
 [description] => string
 [platformid] => string
 [clientid] => string
 [deploymentid] => int
 [urls] =>
 Array
 (
 [icon] => string
 [edit] => string
 [course] => string
 [publickeyset] => string
 [accesstoken] => string
 [authrequest] => string
)
 [state] =>
 Array
 (
 [text] => string
 [pending] => int
 [configured] => int
 [rejected] => int
 [unknown] => int
)
 [hascapabilitygroups] => int
 [capabilitygroups] =>
 Array
 (
 [0] => string
)
 [courseid] => int
 [instanceids] =>
 Array
 (
 [0] => int
)
 [instancecount] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="platformid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="clientid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="deploymentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="urls">
 <SINGLE>
 <KEY name="icon">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="edit">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="publickeyset">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="accesstoken">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authrequest">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="state">
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pending">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="configured">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rejected">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unknown">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="hascapabilitygroups">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="capabilitygroups">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="instanceids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="instancecount">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```
</SINGLE>
</RESPONSE>
```

### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

**mod\_lti\_view\_lti**

Trigger the course module viewed event and update the module completion status.

### Arguments

**ltiid** (Required)

lti instance id

**General structure**

```
int //lti instance id
```

**XML-RPC (PHP structure)**

```
[ltiid] => int
```

**REST (POST parameters)**

```
ltiid= int
```



### Response

**General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

## mod\_page\_get\_pages\_by\_courses

Returns a list of pages in a provided list of courses, if no list is provided all pages that the user can view will be returned.

## Arguments

courseids (Default to "Array ()")

Array of course ids

### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)
```

### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

### REST (POST parameters)

```
courseids[0]= int
```

## Response



**General structure**

```
object {
 pages list of (
 object {
 id int //Activity instance id
 coursemodule int //Course module id
 course int //Course id
 name string //Activity name
 intro string //Activity introduction
 introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 introfiles Optional //Files in the introduction
 list of (
 //File.
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for external files.
 }
)section int Optional //Course section id
 visible int Optional //Visible
 groupmode int Optional //Group mode
 groupingid int Optional //Group id
 lang string Optional //Forced activity language
 content string //Page content
 contentformat int //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 contentfiles //Files in the content
 list of (
 //File.
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for external files.
 }
)legacyfiles int //Legacy files flag
 legacyfileslast int //Legacy files last control flag
 display int //How to display the page
 displayoptions string //Display options (width, height)
 revision int //Incremented when after each file changes, to avoid cache
 timemodified int //Last time the page was modified
)
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



## XML-RPC (PHP structure)

```

Array
(
 [pages] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [content] => string
 [contentformat] => int
 [contentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [legacyfiles] => int
 [legacyfileslast] => int
 [display] => int
 [displayoptions] => string
 [revision] => int
 [timemodified] => int
)
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="filepath">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="legacyfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="legacyfileslast">
 <VALUE>int</VALUE>
</KEY>
<KEY name="display">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displayoptions">
 <VALUE>string</VALUE>
</KEY>
<KEY name="revision">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_page_view_page`

Simulate the view.php web interface page: trigger events, completion, etc...

#### Arguments

`pageid` (Required)

page instance id

##### General structure

```
int //page instance id
```

##### XML-RPC (PHP structure)

```
[pageid] => int
```

##### REST (POST parameters)

```
pageid= int
```

#### Response

##### General structure

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



##### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_get\_attempt\_access\_information**

Return access information for a given attempt in a quiz.

**Arguments**

quizid (Required)

quiz instance id

**General structure**

```
int //quiz instance id
```

**XML-RPC (PHP structure)**

```
[quizid] => int
```

**REST (POST parameters)**

```
quizid= int
```

attemptid (Default to "0")

attempt id, 0 for the user last attempt if exists

**General structure**

```
int Default to "0" //attempt id, 0 for the user last attempt if exists
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**Response****General structure**

```
object {
 endtime int Optional //When the attempt must be submitted (determined by rules).
 isfinished int //Whether there is no way the user will ever be allowed to attempt.
 ispreflightcheckrequired int Optional //whether a check is required before the user
 starts/continues his attempt.
 preventnewattemptreasons //List of reasons
 list of (
 string //access restriction description
)warnings Optional //List of warnings
 list of (
 //warning
)object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [endtime] => int
 [isfinished] => int
 [ispreflightcheckrequired] => int
 [preventnewattemptreasons] =>
 Array
 (
 [0] => string
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="endtime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfinished">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ispreflightcheckrequired">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preventnewattemptreasons">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_get\_attempt\_data**

Returns information for the given attempt page for a quiz attempt in progress.

**Arguments**

attemptid (Required)

attempt id

**General structure**

```
int //attempt id
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**page** (Required)

page number

**General structure**

```
int //page number
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

**General structure**

```
Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

## Response

```

General structure

object {
attempt object {
id int Optional //Attempt id.
quiz int Optional //Foreign key reference to the quiz that was attempted.
userid int Optional //Foreign key reference to the user whose attempt this is.
attempt int Optional //Sequentially numbers this student's attempts at this quiz.
uniqueid int Optional //Foreign key reference to the question_usage that holds the
details of the question_attempts that make up this quiz
attempt.

layout string Optional //Attempt layout.
currentpage int Optional //Attempt current page.
preview int Optional //Whether is a preview attempt or not.
state string Optional //The current state of the attempts. 'inprogress',
'overdue', 'finished' or 'abandoned'.
timestart int Optional //Time when the attempt was started.
timefinish int Optional //Time when the attempt was submitted.
 0 if the attempt has not been submitted yet.
timemodified int Optional //Last modified time.
timemodifiedoffline int Optional //Last modified time via webservices.
timecheckstate int Optional //Next time quiz cron should check attempt for
state changes. NULL means never check.
sumgrades double Optional //Total marks for this attempt.
gradednotificationsenttime int Optional //Time when the student was notified that manual grading of their attempt was complete.
}

messages //access messages, will only be returned for users with mod/quiz:preview capability,
for other users this method will throw an exception if there are messages
list of (
string //access message
)nextpage int //next page number
questions list of (
//The question data. Some fields may not be returned depending on the quiz display settings.
object {
slot int //slot number
type string //question type, i.e. multichoice
page int //page of the quiz this question appears on
questionnumber string //The question number to display for this question, e.g. "7", "i" or "Custom-B").
number int Optional //DO NOT USE. Use questionnumber. Only retained for backwards compatibility.
html string //the question rendered
responsefileareas Optional //Response file areas including files
list of (
object {
area string //File area name
files Optional //Response files for the question
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
})
sequencecheck int Optional //the number of real steps in this attempt
lastactiontime int Optional //the timestamp of the most recent step in this question attempt
hasautosavedstep int Optional //whether this question attempt has autosaved data
flagged int //whether the question is flagged or not
state string Optional //the state where the question is in.
 It will not be returned if the user cannot see it due to the quiz display correctness settings.
status string Optional //current formatted state of the question
blockedbyprevious int Optional //whether the question is blocked by the previous question
mark string Optional //the mark awarded.
 It will be returned only if the user is allowed to see it.
maxmark double Optional //the maximum mark possible for this question attempt.
 It will be returned only if the user is allowed to see it.
settings string Optional //Question settings (JSON encoded).
}

warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}

```



^

**XML-RPC (PHP structure)**

```

Array
(
 [attempt] =>
 Array
 (
 [id] => int
 [quiz] => int
 [userid] => int
 [attempt] => int
 [uniqueid] => int
 [layout] => string
 [currentpage] => int
 [preview] => int
 [state] => string
 [timestart] => int
 [timefinish] => int
 [timemodified] => int
 [timemodifiedoffline] => int
 [timecheckstate] => int
 [sumgrades] => double
 [gradednotificationsenttime] => int
)
 [messages] =>
 Array
 (
 [0] => string
)
 [nextpage] => int
 [questions] =>
 Array
 (
 [0] =>
 Array
 (
 [slot] => int
 [type] => string
 [page] => int
 [questionnumber] => string
 [number] => int
 [html] => string
 [responsefileareas] =>
 Array
 (
 [0] =>
 Array
 (
 [area] => string
 [files] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
 [sequencecheck] => int
 [lastactiontime] => int
 [hasautosavedstep] => int
 [flagged] => int
 [state] => string
 [status] => string
 [blockedbyprevious] => int
 [mark] => string
 [maxmark] => double
 [settings] => string
)
 [warnings] =>
 Array
 (
)
)

```



```
[0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="attempt">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="quiz">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="uniqueid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="currentpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timestart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timefinish">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodifiedoffline">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecheckstate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sumgrades">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradednotificationsenttime">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="messages">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="nextpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="questions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="slot">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="page">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="questionnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="number">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </RESPONSE>
```



```
<KEY name="html">
 <VALUE>string</VALUE>
</KEY>
<KEY name="responsefileareas">
 <MULTIPLE>
 <SINGLE>
 <KEY name="area">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="sequencecheck">
 <VALUE>int</VALUE>
</KEY>
<KEY name="lastactiontime">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasautosavedstep">
 <VALUE>int</VALUE>
</KEY>
<KEY name="flagged">
 <VALUE>int</VALUE>
</KEY>
<KEY name="state">
 <VALUE>string</VALUE>
</KEY>
<KEY name="status">
 <VALUE>string</VALUE>
</KEY>
<KEY name="blockedbyprevious">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mark">
 <VALUE>string</VALUE>
</KEY>
<KEY name="maxmark">
 <VALUE>double</VALUE>
</KEY>
<KEY name="settings">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```

</KEY>
<KEY name="warningcode">
 <VALUE>string</VALUE>
</KEY>
<KEY name="message">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_quiz\_get\_attempt\_review

Returns review information for the given finished attempt, can be used by users or teachers.

#### Arguments

attemptid (Required)

attempt id

#### General structure

```
int //attempt id
```



#### XML-RPC (PHP structure)

```
[attemptid] => int
```

#### REST (POST parameters)

```
attemptid= int
```

page (Default to "-1")

page number, empty for all the questions in all the pages

#### General structure

```
int Default to "-1" //page number, empty for all the questions in all the pages
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

Response



**General structure**

```

object {
grade string //grade for the quiz (or empty or "notyetgraded")
attempt object {
id int Optional //Attempt id.
quiz int Optional //Foreign key reference to the quiz that was attempted.
userid int Optional //Foreign key reference to the user whose attempt this is.
attempt int Optional //Sequentially numbers this student's attempts at this quiz.
uniqueid int Optional //Foreign key reference to the question_usage that holds the
details of the question_attempts that make up this quiz
attempt.

layout string Optional //Attempt layout.
currentpage int Optional //Attempt current page.
preview int Optional //Whether is a preview attempt or not.
state string Optional //The current state of the attempts. 'inprogress',
'overdue', 'finished' or 'abandoned'.
timestart int Optional //Time when the attempt was started.
timefinish int Optional //Time when the attempt was submitted.
 0 if the attempt has not been submitted yet.
timemodified int Optional //Last modified time.
timemodifiedoffline int Optional //Last modified time via webservices.
timecheckstate int Optional //Next time quiz cron should check attempt for
state changes. NULL means never check.
sumgrades double Optional //Total marks for this attempt.
gradednotificationsenttime int Optional //Time when the student was notified that manual grading of their attempt was complete.
}
additionaldata list of (
object {
id string //id of the data
title string //data title
content string //data content
})
questions list of (
//The question data. Some fields may not be returned depending on the quiz display settings.
object {
slot int //slot number
type string //question type, i.e: multichoice
page int //page of the quiz this question appears on
questionnumber string //The question number to display for this question, e.g. "7", "i" or "Custom-B)".
number int Optional //DO NOT USE. Use questionnumber. Only retained for backwards compatibility.
html string //the question rendered
responsefileareas Optional //Response file areas including files
list of (
object {
area string //File area name
files Optional //Response files for the question
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
})
})
sequencecheck int Optional //the number of real steps in this attempt
lastactiontime int Optional //the timestamp of the most recent step in this question attempt
hasautosavedstep int Optional //whether this question attempt has autosaved data
flagged int //whether the question is flagged or not
state string Optional //the state where the question is in.
 It will not be returned if the user cannot see it due to the quiz display correctness settings.
status string Optional //current formatted state of the question
blockedbyprevious int Optional //whether the question is blocked by the previous question
mark string Optional //the mark awarded.
 It will be returned only if the user is allowed to see it.
maxmark double Optional //the maximum mark possible for this question attempt.
 It will be returned only if the user is allowed to see it.
settings string Optional //Question settings (JSON encoded).
)
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
}

```



^

**XML-RPC (PHP structure)**

```

Array
(
 [grade] => string
 [attempt] =>
 Array
 (
 [id] => int
 [quiz] => int
 [userid] => int
 [attempt] => int
 [uniqueid] => int
 [layout] => string
 [currentpage] => int
 [preview] => int
 [state] => string
 [timestart] => int
 [timefinish] => int
 [timemodified] => int
 [timemodifiedoffline] => int
 [timecheckstate] => int
 [sumgrades] => double
 [gradednotificationsenttime] => int
)
 [additionaldata] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => string
 [title] => string
 [content] => string
)
)
)
[questions] =>
 Array
 (
 [0] =>
 Array
 (
 [slot] => int
 [type] => string
 [page] => int
 [questionnumber] => string
 [number] => int
 [html] => string
 [responsefileareas] =>
 Array
 (
 [0] =>
 Array
 (
 [area] => string
 [files] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
)
 [sequencecheck] => int
 [lastactiontime] => int
 [hasautosavedstep] => int
 [flagged] => int
 [state] => string
 [status] => string
 [blockedbyprevious] => int
 [mark] => string
 [maxmark] => double
)

```



```
 [settings] => string
)
)
[warnings] =>
Array
(
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="grade">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="attempt">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="quiz">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="uniqueid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="currentpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timestart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timefinish">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodifiedoffline">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecheckstate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sumgrades">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradednotificationsenttime">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 <KEY name="additionaldata">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="questions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="slot">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</RESPONSE>

```

```
</KEY>
<KEY name="page">
 <VALUE>int</VALUE>
</KEY>
<KEY name="questionnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="number">
 <VALUE>int</VALUE>
</KEY>
<KEY name="html">
 <VALUE>string</VALUE>
</KEY>
<KEY name="responsefileareas">
 <MULTIPLE>
 <SINGLE>
 <KEY name="area">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="sequencecheck">
 <VALUE>int</VALUE>
</KEY>
<KEY name="lastactiontime">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasautosavedstep">
 <VALUE>int</VALUE>
</KEY>
<KEY name="flagged">
 <VALUE>int</VALUE>
</KEY>
<KEY name="state">
 <VALUE>string</VALUE>
</KEY>
<KEY name="status">
 <VALUE>string</VALUE>
</KEY>
<KEY name="blockedbyprevious">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mark">
 <VALUE>string</VALUE>
</KEY>
<KEY name="maxmark">
 <VALUE>double</VALUE>
</KEY>
<KEY name="settings">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
```



```

 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_quiz\_get\_attempt\_summary

Returns a summary of a quiz attempt before it is submitted.



#### Arguments

**attemptid** (Required)

attempt id

#### General structure

```
int //attempt id
```

#### XML-RPC (PHP structure)

```
[attemptid] => int
```

#### REST (POST parameters)

```
attemptid= int
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

**General structure**

```
Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

**Response**

**General structure**

```

object {
questions list of (
 //The question data. Some fields may not be returned depending on the quiz display settings.
object {
slot int //slot number
type string //question type, i.e: multichoice
page int //page of the quiz this question appears on
questionnumber string //The question number to display for this question, e.g. "7", "i" or "Custom-B").
number int Optional //DO NOT USE. Use questionnumber. Only retained for backwards compatibility.
html string //the question rendered
responsefileareas Optional //Response file areas including files
list of (
object {
area string //File area name
files Optional //Response files for the question
list of (
 //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)
})sequencecheck int Optional //the number of real steps in this attempt
lastactiontime int Optional //the timestamp of the most recent step in this question attempt
hasautosavedstep int Optional //whether this question attempt has autosaved data
flagged int //whether the question is flagged or not
state string Optional //the state where the question is in.
 It will not be returned if the user cannot see it due to the quiz display correctness settings.
status string Optional //current formatted state of the question
blockedbyprevious int Optional //whether the question is blocked by the previous question
mark string Optional //the mark awarded.
 It will be returned only if the user is allowed to see it.
maxmark double Optional //the maximum mark possible for this question attempt.
 It will be returned only if the user is allowed to see it.
settings string Optional //Question settings (JSON encoded).
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [questions] =>
 Array
 (
 [0] =>
 Array
 (
 [slot] => int
 [type] => string
 [page] => int
 [questionnumber] => string
 [number] => int
 [html] => string
 [responsefileareas] =>
 Array
 (
 [0] =>
 Array
 (
 [area] => string
 [files] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
 [sequencecheck] => int
 [lastactiontime] => int
 [hasautosavedstep] => int
 [flagged] => int
 [state] => string
 [status] => string
 [blockedbyprevious] => int
 [mark] => string
 [maxmark] => double
 [settings] => string
)
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="questions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="slot">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="page">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="questionnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="number">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="html">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="responsefileareas">
 <MULTIPLE>
 <SINGLE>
 <KEY name="area">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="sequencecheck">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lastactiontime">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasautosavedstep">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flagged">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="blockedbyprevious">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mark">
 <VALUE>string</VALUE>
</KEY>
<KEY name="maxmark">
 <VALUE>double</VALUE>
</KEY>
<KEY name="settings">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

## Error message

### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

## Restricted to logged-in users

Yes



## Callable from AJAX

No

## mod\_quiz\_get\_combined\_review\_options

Combines the review options from a number of different quiz attempts.

### Arguments

quizid (Required)

quiz instance id

### General structure

```
int //quiz instance id
```

**XML-RPC (PHP structure)**

```
[quizid] => int
```

**REST (POST parameters)**

```
quizid= int
```

**userid** (Default to "0")

user id (empty for current user)

**General structure**

```
int Default to "0" //user id (empty for current user)
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response****General structure**

```
object {
 someoptions list of (
 object {
 name string //option name
 value int //option value
 }
)alloptions list of (
 object {
 name string //option name
 value int //option value
 }
)warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [someoptions] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => int
)
)
 [alloptions] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => int
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

^

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="someoptions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="alloptions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_get\_quiz\_access\_information**

Return access information for a given quiz.

## Arguments

quizid (Required)

quiz instance id

### General structure

```
int //quiz instance id
```

### XML-RPC (PHP structure)

```
[quizid] => int
```

### REST (POST parameters)

```
quizid= int
```

## Response

### General structure

```
object {
 canattempt int //Whether the user can do the quiz or not.
 canmanage int //Whether the user can edit the quiz settings or not.
 canpreview int //Whether the user can preview the quiz or not.
 canreviewmyattempts int //Whether the users can review their previous attempts
 //or not.
 canviewreports int //Whether the user can view the quiz reports or not.
 accessrules //list of rules
 list of (
 string //rule description
)activerulenames //list of active rules
 list of (
 string //rule plugin names
)preventaccessreasons //list of reasons
 list of (
 string //access restriction description
)warnings Optional //List of warnings
 list of (
 //warning
)
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [canattempt] => int
 [canmanage] => int
 [canpreview] => int
 [canreviewmyattempts] => int
 [canviewreports] => int
 [accessrules] =>
 Array
 (
 [0] => string
)
 [activerulenames] =>
 Array
 (
 [0] => string
)
 [preventaccessreasons] =>
 Array
 (
 [0] => string
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="canattempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canpreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreviewmyattempts">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewreports">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="accessrules">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="activerulenames">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="preventaccessreasons">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

### mod\_quiz\_get\_quiz\_feedback\_for\_grade

Get the feedback text that should be show to a student who got the given grade in the given quiz.

#### Arguments

**quizid** (Required)

quiz instance id

##### General structure

```
int //quiz instance id
```

##### XML-RPC (PHP structure)

```
[quizid] => int
```

##### REST (POST parameters)

```
quizid= int
```

**grade** (Required)

the grade to check

##### General structure

```
double //the grade to check
```

##### XML-RPC (PHP structure)

```
[grade] => double
```

##### REST (POST parameters)

```
grade= double
```

#### Response



## General structure

```
object {
feedbacktext string //the comment that corresponds to this grade (empty for none)
feedbacktextformat int Optional //feedbacktext format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
feedbackinlinefiles Optional //feedback inline files
list of (
 //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
})}
```

## XML-RPC (PHP structure)

```
 Array
 (
 [feedbacktext] => string
 [feedbacktextformat] => int
 [feedbackinlinelinefiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="feedbacktext">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbacktextformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackinlinelinefiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

## REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

#### Callable from AJAX

No

#### mod\_quiz\_get\_quiz\_required\_qtypes

Return the potential question types that would be required for a given quiz.

#### Arguments

**quizid** (Required)

quiz instance id

##### General structure

```
int //quiz instance id
```

##### XML-RPC (PHP structure)

```
[quizid] => int
```

##### REST (POST parameters)

```
quizid= int
```

#### Response

##### General structure

```
object {
 questiontypes //list of question types used in the quiz
 list of (
 string //question type
)warnings Optional //list of warnings
 list of (
 //warning
)object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [questiontypes] =>
 Array
 (
 [
 [0] => string
]
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="questiontypes">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

## mod\_quiz\_get\_quizzes\_by\_courses

Returns a list of quizzes in a provided list of courses, if no list is provided all quizzes that the user can view will be returned.

### Arguments

courseids (Default to "Array ()")

Array of course ids

#### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

#### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
)
```

#### REST (POST parameters)

```
courseids[0]= int
```

### Response



**General structure**

```

object {
 quizzes list of (
 object {
 id int //Activity instance id
 coursemodule int //Course module id
 course int //Course id
 name string //Activity name
 intro string Optional //Activity introduction
 introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 introfiles Optional //Files in the introduction
 list of (
 //File.
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for external files.
 }
)section int Optional //Course section id
 visible int Optional //Visible
 groupmode int Optional //Group mode
 groupingid int Optional //Group id
 lang string Optional //Forced activity language
 timeopen int Optional //The time when this quiz opens. (0 = no restriction.)
 timeclose int Optional //The time when this quiz closes. (0 = no restriction.)
 timelimit int Optional //The time limit for quiz attempts, in seconds.
 overduehandling string Optional //The method used to handle overdue attempts.
 'autosubmit', 'graceperiod' or 'autoabandon'.
 graceperiod int Optional //The amount of time (in seconds) after the time limit
 runs out during which attempts can still be submitted,
 if overduehandling is set to allow it.
 preferredbehaviour string Optional //The behaviour to ask questions to use.
 canredoquestions int Optional //Allows students to redo any completed question
 within a quiz attempt.
 attempts int Optional //The maximum number of attempts a student is allowed.
 attemptonlast int Optional //Whether subsequent attempts start from the answer
 to the previous attempt (1) or start blank (0).
 grademethod int Optional //One of the values QUIZ_GRADEHIGHEST, QUIZ_GRADEAVERAGE,
 QUIZ_ATTEMPTFIRST or QUIZ_ATTEMPTLAST.
 decimalpoints int Optional //Number of decimal points to use when displaying
 grades.
 questiondecimalpoints int Optional //Number of decimal points to use when
 displaying question grades.
 (-1 means use decimalpoints.)
 reviewattempt int Optional //Whether users are allowed to review their quiz
 attempts at various times. This is a bit field, decoded by the
 \mod_quiz\question\display_options class. It is formed by ORing
 together the constants defined there.
 reviewcorrectness int Optional //Whether users are allowed to review their quiz
 attempts at various times.
 reviewmarks int Optional //Whether users are allowed to review their quiz attempts
 at various times. A bit field, like reviewattempt.
 reviewspecificfeedback int Optional //Whether users are allowed to review their
 quiz attempts at various times. A bit field, like
 reviewattempt.
 reviewgeneralfeedback int Optional //Whether users are allowed to review their
 quiz attempts at various times. A bit field, like
 reviewattempt.
 reviewrightanswer int Optional //Whether users are allowed to review their quiz
 attempts at various times. A bit field, like
 reviewattempt.
 reviewoverallfeedback int Optional //Whether users are allowed to review their quiz
 attempts at various times. A bit field, like
 reviewattempt.
 questionsperpage int Optional //How often to insert a page break when editing
 the quiz, or when shuffling the question order.
 navmethod string Optional //Any constraints on how the user is allowed to navigate
 around the quiz. Currently recognised values are
 'free' and 'seq'.
 shuffleanswers int Optional //Whether the parts of the question should be shuffled,
 in those question types that support it.
 sumgrades double Optional //The total of all the question instance maxmarks.
 grade double Optional //The total that the quiz overall grade is scaled to be
 out of.
 timecreated int Optional //The time when the quiz was added to the course.
 timemodified int Optional //Last modified time.
 }
)
}

```

```
password string Optional //A password that the student must enter before starting or
 continuing a quiz attempt.
subnet string Optional //Used to restrict the IP addresses from which this quiz can
 be attempted. The format is as required by the address_in_subnet
 function.
browsersecurity string Optional //Restriction on the browser the student must
 use. E.g. 'securewindow'.
delay1 int Optional //Delay that must be left between the first and second attempt,
 in seconds.
delay2 int Optional //Delay that must be left between the second and subsequent
 attempt, in seconds.
showuserpicture int Optional //Option to show the user's picture during the
 attempt and on the review page.
showblocks int Optional //Whether blocks should be shown on the attempt.php and
 review.php pages.
completionattemptsexhausted int Optional //Mark quiz complete when the student has
 exhausted the maximum number of attempts
completionpass int Optional //Whether to require passing grade
allowofflineattempts int Optional //Whether to allow the quiz to be attempted
 offline in the mobile app
autosaveperiod int Optional //Auto-save delay
hasfeedback int Optional //Whether the quiz has any non-blank feedback text
hasquestions int Optional //Whether the quiz has questions
}
)warnings Optional //List of warnings
list of (
 /warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})
```



**XML-RPC (PHP structure)**

```

Array
(
 [quizzes] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [timeopen] => int
 [timeclose] => int
 [timelimit] => int
 [overduehandling] => string
 [graceperiod] => int
 [preferredbehaviour] => string
 [canredoquestions] => int
 [attempts] => int
 [attemptonlast] => int
 [grademethod] => int
 [decimalpoints] => int
 [questiondecimalpoints] => int
 [reviewattempts] => int
 [reviewcorrectness] => int
 [reviewmarks] => int
 [reviewspecificfeedback] => int
 [reviewgeneralfeedback] => int
 [reviewrightanswer] => int
 [reviewoverallfeedback] => int
 [questionsperpage] => int
 [navmethod] => string
 [shuffleanswers] => int
 [sumgrades] => double
 [grade] => double
 [timecreated] => int
 [timemodified] => int
 [password] => string
 [subnet] => string
 [browsersecurity] => string
 [delay1] => int
 [delay2] => int
 [showuserpicture] => int
 [showblocks] => int
 [completionattemptsexhausted] => int
 [completionpass] => int
 [allowofflineattempts] => int
 [autosaveperiod] => int
 [hasfeedback] => int
 [hasquestions] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>

```



```
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="quizzes">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timeopen">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeclose">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timelimit">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="overduehandling">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="graceperiod">
 <VALUE>int</VALUE>
</KEY>
<KEY name="preferredbehaviour">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canredoquestions">
 <VALUE>int</VALUE>
</KEY>
<KEY name="attempts">
 <VALUE>int</VALUE>
</KEY>
<KEY name="attemptonlast">
 <VALUE>int</VALUE>
</KEY>
<KEY name="grademethod">
 <VALUE>int</VALUE>
</KEY>
<KEY name="decimalpoints">
 <VALUE>int</VALUE>
</KEY>
<KEY name="questiondecimalpoints">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewattempt">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewcorrectness">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewmarks">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewspecificfeedback">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewgeneralfeedback">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewrightanswer">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewoverallfeedback">
 <VALUE>int</VALUE>
</KEY>
<KEY name="questionsperpage">
 <VALUE>int</VALUE>
</KEY>
<KEY name="navmethod">
 <VALUE>string</VALUE>
</KEY>
<KEY name="shuffleanswers">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sumgrades">
 <VALUE>double</VALUE>
</KEY>
<KEY name="grade">
 <VALUE>double</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="password">
 <VALUE>string</VALUE>
</KEY>
<KEY name="subnet">
 <VALUE>string</VALUE>
</KEY>
<KEY name="browsersecurity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="delay1">
 <VALUE>int</VALUE>
</KEY>
<KEY name="delay2">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showuserpicture">
 <VALUE>int</VALUE>
```



```

</KEY>
<KEY name="showblocks">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completionattemptsexhausted">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completionpass">
 <VALUE>int</VALUE>
</KEY>
<KEY name="allowofflineattempts">
 <VALUE>int</VALUE>
</KEY>
<KEY name="autosaveperiod">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasfeedback">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasquestions">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>

```

### Error message

**REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_quiz\_get\_reopen\_attempt\_confirmation

Verify it is OK to re-open a given quiz attempt, and if so, return a suitable confirmation message.

### Arguments

**attemptid** (Required)

The id of the attempt to reopen

**General structure**

```
int //The id of the attempt to reopen
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**Response**

Confirmation to show the user before the attempt is reopened.

**General structure**

```
string //Confirmation to show the user before the attempt is reopened.
```

**XML-RPC (PHP structure)**

```
string
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <VALUE>string</VALUE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_quiz\_get\_user\_attempts**

Return a list of attempts for the given quiz and user.

**Arguments**

quizid (Required)

quiz instance id

**General structure**

```
int //quiz instance id
```

**XML-RPC (PHP structure)**

```
[quizid] => int
```

**REST (POST parameters)**

```
quizid= int
```

userid (Default to "0")

user id, empty for current user

**General structure**

```
int Default to "0" //user id, empty for current user
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

status (Default to "finished")

quiz status: all, finished or unfinished

**General structure**

```
string Default to "finished" //quiz status: all, finished or unfinished
```

**XML-RPC (PHP structure)**

```
[status] => string
```

**REST (POST parameters)**

```
status= string
```

includepreviews (Default to "")

whether to include previews or not

**General structure**

```
int Default to "" //whether to include previews or not
```

**XML-RPC (PHP structure)**

```
[includepreviews] => int
```

**REST (POST parameters)**

```
includepreviews= int
```

## Response

```
General structure

object {
 attempts list of (
 object {
 id int Optional //Attempt id.
 quiz int Optional //Foreign key reference to the quiz that was attempted.
 userid int Optional //Foreign key reference to the user whose attempt this is.
 attempt int Optional //Sequentially numbers this student's attempts at this quiz.
 uniqueid int Optional //Foreign key reference to the question_usage that holds the
 details of the question_attempts that make up this quiz
 attempt.

 layout string Optional //Attempt layout.
 currentpage int Optional //Attempt current page.
 preview int Optional //Whether is a preview attempt or not.
 state string Optional //The current state of the attempts. 'inprogress',
 'overdue', 'finished' or 'abandoned'.
 timestamp int Optional //Time when the attempt was started.
 timefinish int Optional //Time when the attempt was submitted.
 0 if the attempt has not been submitted yet.
 timemodified int Optional //Last modified time.
 timemodifiedoffline int Optional //Last modified time via webservices.
 timecheckstate int Optional //Next time quiz cron should check attempt for
 state changes. NULL means never check.
 sumgrades double Optional //Total marks for this attempt.
 gradednotificationsendtime int Optional //Time when the student was notified that manual grading of their attempt was complete.
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [attempts] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [quiz] => int
 [userid] => int
 [attempt] => int
 [uniqueid] => int
 [layout] => string
 [currentpage] => int
 [preview] => int
 [state] => string
 [timestart] => int
 [timefinish] => int
 [timemodified] => int
 [timemodifiedoffline] => int
 [timecheckstate] => int
 [sumgrades] => double
 [gradednotificationsenttime] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="attempts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="quiz">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="uniqueid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="currentpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timestart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timefinish">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodifiedoffline">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecheckstate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sumgrades">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradednotificationsenttime">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```



## Error message

### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

No

## mod\_quiz\_get\_user\_best\_grade

Get the best current grade for the given user on a quiz.

### Arguments

**quizid** (Required)

quiz instance id

#### General structure

```
int //quiz instance id
```

#### XML-RPC (PHP structure)

```
[quizid] => int
```

#### REST (POST parameters)

```
quizid= int
```

**userid** (Default to "0")

user id



#### General structure

```
int Default to "0" //user id
```

#### XML-RPC (PHP structure)

```
[userid] => int
```

#### REST (POST parameters)

```
userid= int
```

## Response

## General structure

```
object {
hasgrade int //Whether the user has a grade on the given quiz.
grade double Optional //The grade (only if the user has a grade).
gradetopass double Optional //The grade to pass the quiz (only if set).
warnings Optional //list of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
})}
```

## XML-RPC (PHP structure)

```
Array
(
 [hasgrade] => int
 [grade] => double
 [gradetopass] => double
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="hasgrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradetopass">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_process\_attempt**

Process responses during an attempt at a quiz and also deals with attempts finishing.

**Arguments****attemptid** (Required)

attempt id

**General structure**

```
int //attempt id
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**data** (Default to "Array ()")

the data to be saved

**General structure**

```
Default to "Array
(
)
" //the data to be saved
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[data] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
```

**REST (POST parameters)**

```
data[0][name]= string
data[0][value]= string
```

**finishattempt** (Default to "")

whether to finish or not the attempt

**General structure**

```
int Default to "" //whether to finish or not the attempt
```

**XML-RPC (PHP structure)**

```
[finishattempt] => int
```

**REST (POST parameters)**

```
finishattempt= int
```

**timeup** (Default to "")

whether the WS was called by a timer when the time is up

**General structure**

```
int Default to "" //whether the WS was called by a timer when the time is up
```

**XML-RPC (PHP structure)**

```
[timeup] => int
```

**REST (POST parameters)**

```
timeup= int
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

**General structure**

```
Default to "Array"
(
)
" //Preflight required data (Like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

**Response****General structure**

```
object {
 state string //state: the new attempt state: inprogress, finished, overdue, abandoned
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [state] => string
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_reopen\_attempt**

Re-open an attempt that is currently in the never submitted state.

**Arguments****attemptid** (Required)

The id of the attempt to reopen

**General structure**

```
int //The id of the attempt to reopen
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**Response****Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_quiz\_save\_attempt**

Processes save requests during the quiz. This function is intended for the quiz auto-save feature.

**Arguments**

**attemptid** (Required)

attempt id

**General structure**

```
int //attempt id
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**data** (Required)

the data to be saved

**General structure**

```
//the data to be saved
list of (
object {
name string //data name
value string //data value
})
```

**XML-RPC (PHP structure)**

```
[data] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
 ^
```

**REST (POST parameters)**

```
data[0][name]= string
data[0][value]= string
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

```
General structure

Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

**Response**

```
General structure

object {
status int //status: true if success
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_set\_question\_version**

Set the version of question that would be required for a given quiz.

**Arguments****slotid** (Required)**General structure**

int

**XML-RPC (PHP structure)**

[slotid] =&gt; int

**REST (POST parameters)**

```
slotid= int
```

newversion (Required)

**General structure**

```
int
```

**XML-RPC (PHP structure)**

```
[newversion] => int
```

**REST (POST parameters)**

```
newversion= int
```

**Response****General structure**

```
object {
 result int }
```

**XML-RPC (PHP structure)**

```
Array
(
 [result] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**mod\_quiz\_start\_attempt**

Starts a new attempt at a quiz.

**Arguments**

**quizid** (Required)

quiz instance id

**General structure**

```
int //quiz instance id
```

**XML-RPC (PHP structure)**

```
[quizid] => int
```

**REST (POST parameters)**

```
quizid= int
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

**General structure**

```
Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

**forcenew** (Default to "")

Whether to force a new attempt or not.

**General structure**

```
int Default to "" //Whether to force a new attempt or not.
```

**XML-RPC (PHP structure)**

```
[forcenew] => int
```

**REST (POST parameters)**

```
forcenew= int
```

**Response****General structure**

```
object {
attempt object {
id int Optional //Attempt id.
quiz int Optional //Foreign key reference to the quiz that was attempted.
userid int Optional //Foreign key reference to the user whose attempt this is.
attempt int Optional //Sequentially numbers this student's attempts at this quiz.
uniqueid int Optional //Foreign key reference to the question_usage that holds the
details of the the question_attempts that make up this quiz
attempt.

layout string Optional //Attempt layout.
currentpage int Optional //Attempt current page.
preview int Optional //Whether is a preview attempt or not.
state string Optional //The current state of the attempts. 'inprogress',
'overdue', 'finished' or 'abandoned'.
timestart int Optional //Time when the attempt was started.
timefinish int Optional //Time when the attempt was submitted.
 0 if the attempt has not been submitted yet.
timemodified int Optional //Last modified time.
timemodifiedoffline int Optional //Last modified time via webservices.
timecheckstate int Optional //Next time quiz cron should check attempt for
state changes. NULL means never check.
sumgrades double Optional //Total marks for this attempt.
gradednotificationsendtime int Optional //Time when the student was notified that manual grading of their attempt was complete.
}
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [attempt] =>
 Array
 (
 [id] => int
 [quiz] => int
 [userid] => int
 [attempt] => int
 [uniqueid] => int
 [layout] => string
 [currentpage] => int
 [preview] => int
 [state] => string
 [timestart] => int
 [timefinish] => int
 [timemodified] => int
 [timemodifiedoffline] => int
 [timecheckstate] => int
 [sumgrades] => double
 [gradednotificationsenttime] => int
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="attempt">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="quiz">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="uniqueid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="layout">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="currentpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="preview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="state">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timestart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timefinish">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodifiedoffline">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecheckstate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sumgrades">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradednotificationsenttime">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_view\_attempt**

Trigger the attempt viewed event.

**Arguments****attemptid** (Required)

attempt id

**General structure**

```
int //attempt id
```

**XML-RPC (PHP structure)**

```
[attemptid] => int
```

**REST (POST parameters)**

```
attemptid= int
```

**page** (Required)

page number

^

**General structure**

```
int //page number
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

```
General structure

Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

## XML-RPC (PHP structure)

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
)
```

## REST (POST parameters)

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

## Response

```
General structure

object {
status int //status: true if success
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```

## XML-RPC (PHP structure)

```
Array
(
[status] => int
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_view\_attempt\_review**

Trigger the attempt reviewed event.

**Arguments**

attemptid (Required)

attempt id

**General structure**

int //attempt id

**XML-RPC (PHP structure)**

[attemptid] =&gt; int

**REST (POST parameters)**

```
attemptid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_view\_attempt\_summary**

Trigger the attempt summary viewed event.

**Arguments****attemptid** (Required)

attempt id

**General structure**

int //attempt id

**XML-RPC (PHP structure)**

[attemptid] =&gt; int

**REST (POST parameters)**

attemptid= int

**preflightdata** (Default to "Array ()")

Preflight required data (like passwords)

**General structure**

```
Default to "Array
(
)
" //Preflight required data (like passwords)
list of (
object {
name string //data name
value string //data value
}
)
```

**XML-RPC (PHP structure)**

```
[preflightdata] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
```

**REST (POST parameters)**

```
preflightdata[0][name]= string
preflightdata[0][value]= string
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_quiz\_view\_quiz**

Trigger the course module viewed event and update the module completion status.

**Arguments****quizid** (Required)

quiz instance id

**General structure**

```
int //quiz instance id
```

**XML-RPC (PHP structure)**

```
[quizid] => int
```

**REST (POST parameters)**

```
quizid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [status] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

No

### **mod\_resource\_get\_resources\_by\_courses**

Returns a list of files in a provided list of courses, if no list is provided all files that the user can view will be returned.

## Arguments

courseids (Default to "Array ()")

Array of course ids

### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)
```

### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

### REST (POST parameters)

```
courseids[0]= int
```

## Response



```
General structure

object {
resources list of (
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
contentfiles //Files in the content
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)to bemigrated int //Whether this resource was migrated
legacyfiles int //Legacy files flag
legacyfileslast int //Legacy files last control flag
display int //How to display the resource
displayoptions string //Display options (width, height)
filterfiles int //If filters should be applied to the resource content
revision int //Incremented when after each file changes, to avoid cache
timemodified int //Last time the resource was modified
}
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [resources] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [contentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [
 [tobemigrated] => int
 [legacyfiles] => int
 [legacyfileslast] => int
 [display] => int
 [displayoptions] => string
 [filterfiles] => int
 [revision] => int
 [timemodified] => int
)
)
)
)
)
)
)
]
)
)
]
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="resources">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="tobemigrated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="legacyfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="legacyfileslast">
 <VALUE>int</VALUE>
</KEY>
<KEY name="display">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displayoptions">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filterfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="revision">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_resource_view_resource`

Simulate the view.php web interface resource: trigger events, completion, etc...

#### Arguments

`resourceid` (Required)

resource instance id

##### General structure

```
int //resource instance id
```

##### XML-RPC (PHP structure)

```
[resourceid] => int
```

##### REST (POST parameters)

```
resourceid= int
```

#### Response

##### General structure

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



##### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_scorm\_get\_scorm\_access\_information**

Return capabilities information for a given scorm.

**Arguments****scormid** (Required)

scorm instance id.

**General structure**

```
int //scorm instance id.
```

**XML-RPC (PHP structure)**

```
[scormid] => int
```

## REST (POST parameters)

```
scormid= int
```

## Response

## General structure

```
object {
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)canaddinstance int Optional //Whether the user has the capability mod/scorm:addinstance allowed.
 canviewreport int Optional //Whether the user has the capability mod/scorm:viewreport allowed.
 canskipview int Optional //Whether the user has the capability mod/scorm:skipview allowed.
 cansavetrack int Optional //Whether the user has the capability mod/scorm:savetrack allowed.
 canviewscores int Optional //Whether the user has the capability mod/scorm:viewscores allowed.
 candeleteresponses int Optional //Whether the user has the capability mod/scorm:deleteresponses allowed.
 candeleteownresponses int Optional //Whether the user has the capability mod/scorm:deleteownresponses allowed.
}
```

## XML-RPC (PHP structure)

```
Array
(
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
 [canaddinstance] => int
 [canviewreport] => int
 [canskipview] => int
 [cansavetrack] => int
 [canviewscores] => int
 [candeleteresponses] => int
 [candeleteownresponses] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="canaddinstance">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewreport">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canskipview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cansavetrack">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewscores">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="candeleteresponses">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="candeleteownresponses">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

## REST



```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

No

## mod\_scorm\_get\_scorm\_attempt\_count

Return the number of attempts done by a user in the given SCORM.

**Arguments****scormid** (Required)

SCORM instance id

**General structure**

```
int //SCORM instance id
```

**XML-RPC (PHP structure)**

```
[scormid] => int
```

**REST (POST parameters)**

```
scormid= int
```

**userid** (Required)

User id

**General structure**

```
int //User id
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**ignoremissingcompletion** (Default to "")

Ignores attempts that haven't reported a grade/completion

**General structure**

```
int Default to "" //Ignores attempts that haven't reported a grade/completion
```

**XML-RPC (PHP structure)**

```
[ignoremissingcompletion] => int
```

**REST (POST parameters)**

```
ignoremissingcompletion= int
```

**Response**

**General structure**

```

object {
attemptscount int //Attempts count
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
)

```

**XML-RPC (PHP structure)**

```

Array
(
 [
[attemptscount] => int
[warnings] =>
 Array
 (
 [
@0] =>
 Array
 (
 [
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="attemptscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

`mod_scorm_get_scorm_sco_tracks`

Retrieves SCO tracking data for the given user id and attempt number

#### Arguments

`scoid` (Required)

sco id

##### General structure

```
int //sco id
```

##### XML-RPC (PHP structure)

```
[scoid] => int
```

##### REST (POST parameters)

```
scoid= int
```

`userid` (Required)

user id

##### General structure

```
int //user id
```

##### XML-RPC (PHP structure)

```
[userid] => int
```

##### REST (POST parameters)

```
userid= int
```

`attempt` (Default to "0")

attempt number (0 for last attempt)

##### General structure

```
int Default to "0" //attempt number (0 for Last attempt)
```

##### XML-RPC (PHP structure)

```
[attempt] => int
```

##### REST (POST parameters)

```
attempt= int
```

**Response**

```
General structure

object {
 data //SCO data
 object {
 attempt int //Attempt number
 tracks list of (
 //Tracks data
 object {
 element string //Element name
 value string //Element value
 }
)
 }
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [
 [data] =>
 Array
 (
 [
 [attempt] => int
 [tracks] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [element] => string
 [value] => string
)
)
)
)
)
 [
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="data">
 <SINGLE>
 <KEY name="attempt">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tracks">
 <MULTIPLE>
 <SINGLE>
 <KEY name="element">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_scorm\_get\_scorm\_scoes**

Returns a list containing all the scoes data related to the given scorm id

**Arguments**

scormid (Required)

scorm instance id

**General structure**

```
int //scorm instance id
```

**XML-RPC (PHP structure)**

```
[scormid] => int
```

**REST (POST parameters)**

```
scormid= int
```

organization (Default to "")

organization id

**General structure**

```
string Default to "" //organization id
```

**XML-RPC (PHP structure)**

```
[organization] => string
```

**REST (POST parameters)**

```
organization= string
```

**Response****General structure**

```
object {
 scoes list of (
 //SCORM SCO data
 object {
 id int //sco id
 scorm int //scorm id
 manifest string //manifest id
 organization string //organization id
 parent string //parent
 identifier string //identifier
 launch string //Launch file
 scormtype string //scorm type (asset, sco)
 title string //sco title
 sortorder int //sort order
 extradata Optional //Additional SCO data
 }
 list of (
 object {
 element string //element name
 value string //element value
 }
)
 }
 warnings Optional //list of warnings
 list of (
 //warning
)
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [
 [scoes] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [scorm] => int
 [manifest] => string
 [organization] => string
 [parent] => string
 [identifier] => string
 [launch] => string
 [scormtype] => string
 [title] => string
 [sortorder] => int
 [extradata] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [element] => string
 [value] => string
)
)
)
)
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="scoes">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scorm">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="manifest">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="organization">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="parent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identifier">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="launch">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scormtype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="extradata">
 <MULTIPLE>
 <SINGLE>
 <KEY name="element">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_scorm\_get\_scorm\_user\_data**

Retrieves user tracking and SCO data and default SCORM values

**Arguments****scormid** (Required)

scorm instance id

**General structure**

```
int //scorm instance id
```

**XML-RPC (PHP structure)**

```
[scormid] => int
```

**REST (POST parameters)**

```
scormid= int
```

**attempt** (Required)

attempt number

**General structure**

```
int //attempt number
```

**XML-RPC (PHP structure)**

```
[attempt] => int
```

**REST (POST parameters)**

```
attempt= int
```

**Response**

**General structure**

```

object {
 data list of (
 //SCO data
 object {
 scoid int //sco id
 userdata list of (
 object {
 element string //element name
 value string //element value
 }
)defaultdata list of (
 object {
 element string //element name
 value string //element value
 }
)
)warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [data] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [scoid] => int
 [userdata] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [element] => string
 [value] => string
)
)
)
)
)
)
)
 [defaultdata] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [element] => string
 [value] => string
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="data">
 <MULTIPLE>
 <SINGLE>
 <KEY name="scoid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userdata">
 <MULTIPLE>
 <SINGLE>
 <KEY name="element">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="defaultdata">
 <MULTIPLE>
 <SINGLE>
 <KEY name="element">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

## REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

**Callable from AJAX**

No

**mod\_scorm\_get\_scorms\_by\_courses**

Returns a list of scorm instances in a provided set of courses, if no courses are provided then all the scorm instances the user has access to will be returned.

**Arguments****courseids** (Default to "Array ()")

Array of course ids

**General structure**

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [0] => int
)
)
```

**REST (POST parameters)**

courseids[0]= int

**Response**

**General structure**

```

object {
scorms list of (
 //SCORM
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string //Activity introduction
introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
packagesize int Optional //SCORM zip package size
packageurl string Optional //SCORM zip package URL
version string Optional //SCORM version (SCORM_12, SCORM_13, SCORM_AICC)
maxgrade int Optional //Max grade
grademethod int Optional //Grade method
whatgrade int Optional //What grade
maxattempt int Optional //Maximum number of attempts
forcecompleted int Optional //Status current attempt is forced to "completed"
forcenewattempt int Optional //Controls re-entry behaviour
lastattemptlock int Optional //Prevents to launch new attempts once finished
displayattemptstatus int Optional //How to display attempt status
displaycoursestructure int Optional //Display contents structure
sha1hash string Optional //Package content or ext path hash
md5hash string Optional //MD5 Hash of package file
revision int Optional //Revision number
launch int Optional //First content to launch
skipview int Optional //How to skip the content structure page
hidebrowse int Optional //Disable preview mode?
hidetoc int Optional //How to display the SCORM structure in player
nav int Optional //Show navigation buttons
navpositionleft int Optional //Navigation position left
navpositiontop int Optional //Navigation position top
auto int Optional //Auto continue?
popup int Optional //Display in current or new window
width int Optional //Frame width
height int Optional //Frame height
timeopen int Optional //Available from
timeclose int Optional //Available to
scormtype string Optional //SCORM type
reference string Optional //Reference to the package
protectpackagedownloads int Optional //Protect package downloads?
updatefreq int Optional //Auto-update frequency for remote packages
options string Optional //Additional options
completionstatusrequired int Optional //Status passed/completed required?
completionscorerequired int Optional //Minimum score required
completionstatusallscos int Optional //Require all scos to return completion status
autocommit int Optional //Save track data automatically?
timemodified int Optional //Time of last modification
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [scorms] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [packagesize] => int
 [packageurl] => string
 [version] => string
 [maxgrade] => int
 [grademethod] => int
 [whatgrade] => int
 [maxattempt] => int
 [forcecompleted] => int
 [forcenewattempt] => int
 [lastattemptlock] => int
 [displayattemptstatus] => int
 [displaycoursestructure] => int
 [sha1hash] => string
 [md5hash] => string
 [revision] => int
 [launch] => int
 [skipview] => int
 [hidebrowse] => int
 [hidetoc] => int
 [nav] => int
 [navpositionleft] => int
 [navpositiontop] => int
 [auto] => int
 [popup] => int
 [width] => int
 [height] => int
 [timeopen] => int
 [timeclose] => int
 [scormtype] => string
 [reference] => string
 [protectpackagedownloads] => int
 [updatefreq] => int
 [options] => string
 [completionstatusrequired] => int
 [completionscorerequired] => int
 [completionstatusallskos] => int
 [autocommit] => int
 [timemodified] => int
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array

```

^

```
(
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="scorms">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="packagesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="packageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="version">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="maxgrade">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="grademethod">
 <VALUE>int</VALUE>
</KEY>
<KEY name="whatgrade">
 <VALUE>int</VALUE>
</KEY>
<KEY name="maxattempt">
 <VALUE>int</VALUE>
</KEY>
<KEY name="forcecompleted">
 <VALUE>int</VALUE>
</KEY>
<KEY name="forcenewattempt">
 <VALUE>int</VALUE>
</KEY>
<KEY name="lastattemptlock">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displayattemptstatus">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycoursestructure">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sha1hash">
 <VALUE>string</VALUE>
</KEY>
<KEY name="md5hash">
 <VALUE>string</VALUE>
</KEY>
<KEY name="revision">
 <VALUE>int</VALUE>
</KEY>
<KEY name="launch">
 <VALUE>int</VALUE>
</KEY>
<KEY name="skipview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidebrowse">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidetoc">
 <VALUE>int</VALUE>
</KEY>
<KEY name="nav">
 <VALUE>int</VALUE>
</KEY>
<KEY name="navpositionleft">
 <VALUE>int</VALUE>
</KEY>
<KEY name="navpositiontop">
 <VALUE>int</VALUE>
</KEY>
<KEY name="auto">
 <VALUE>int</VALUE>
</KEY>
<KEY name="popup">
 <VALUE>int</VALUE>
</KEY>
<KEY name="width">
 <VALUE>int</VALUE>
</KEY>
<KEY name="height">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeopen">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeclose">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scormtype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="reference">
 <VALUE>string</VALUE>
</KEY>
<KEY name="protectpackagedownloads">
 <VALUE>int</VALUE>
</KEY>
<KEY name="updatefreq">
 <VALUE>int</VALUE>
```



```

 </KEY>
 <KEY name="options">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="completionstatusrequired">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completionscorerequired">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completionstatusallscores">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="autocommit">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



### Restricted to logged-in users

Yes

### Callable from AJAX

No

### `mod_scorm_insert_scorm_tracks`

Saves a scorm tracking record. It will overwrite any existing tracking data for this attempt. Validation should be performed before running the function to ensure the user will not lose any existing attempt data.

### Arguments

`scoid` (Required)

SCO id

**General structure**

```
int //SCO id
```

**XML-RPC (PHP structure)**

```
[scoid] => int
```

**REST (POST parameters)**

```
scoid= int
```

attempt (Required)

attempt number

**General structure**

```
int //attempt number
```

**XML-RPC (PHP structure)**

```
[attempt] => int
```

**REST (POST parameters)**

```
attempt= int
```

tracks (Required)

**General structure**

```
list of (
object {
element string //element name
value string //element value
}
)
```

**XML-RPC (PHP structure)**

```
[tracks] =>
 Array
 (
 [0] =>
 Array
 (
 [element] => string
 [value] => string
)
)
)
```

**REST (POST parameters)**

```
tracks[0][element]= string
tracks[0][value]= string
```

**Response**

## General structure

```
object {
trackids list of (
int //track id
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}
```

## XML-RPC (PHP structure)

```
Array
(
 [trackkids] =>
 Array
 (
 [0] => int
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="trackids">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_scorm\_launch\_sco**

Trigger the SCO launched event.

**Arguments****scormid** (Required)

SCORM instance id

**General structure**

```
int //SCORM instance id
```

**XML-RPC (PHP structure)**

```
[scormid] => int
```

**REST (POST parameters)**

```
scormid= int
```

**scoid** (Default to "0")

SCO id (empty for launching the first SCO)

**General structure**

```
int Default to "0" //SCO id (empty for launching the first SCO)
```

**XML-RPC (PHP structure)**

```
[scoid] => int
```

**REST (POST parameters)**

```
scoid= int
```

**Response**

**General structure**

```

object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

mod\_scorm\_view\_scorm

Trigger the course module viewed event.

#### Arguments

scormid (Required)

scorm instance id

##### General structure

```
int //scorm instance id
```

##### XML-RPC (PHP structure)

```
[scormid] => int
```

##### REST (POST parameters)

```
scormid= int
```

#### Response

##### General structure

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



##### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_survey\_get\_questions**

Get the complete list of questions for the survey, including subquestions.

**Arguments****surveyid** (Required)

survey instance id

**General structure**

```
int //survey instance id
```

**XML-RPC (PHP structure)**

```
[surveyid] => int
```

**REST (POST parameters)**

```
surveyid= int
```

**Response****General structure**

```
object {
 questions list of (
 //Questions
 object {
 id int //Question id
 text string //Question text
 shorttext string //Question short text
 multi string //Subquestions ids
 intro string //The question intro
 type int //Question type
 options string //Question options
 parent int //Parent question (for subquestions)
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [questions] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [text] => string
 [shorttext] => string
 [multi] => string
 [intro] => string
 [type] => int
 [options] => string
 [parent] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="questions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shorttext">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="multi">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="options">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="parent">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

## REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

No

### mod\_survey\_get\_surveys\_by\_courses

Returns a list of survey instances in a provided set of courses, if no courses are provided then all the survey instances the user has access to will be returned.

#### Arguments

courseids (Default to 'Array ()')

Array of course ids

#### General structure

```
Default to "Array
(
)
" //Array of course ids
list of (
int //course id
)
```

#### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

#### REST (POST parameters)

```
courseids[0]= int
```

#### Response



**General structure**

```
object {
surveys list of (
 //Surveys
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string Optional //Activity introduction
introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
template int Optional //Survey type
days int Optional //Days
questions string Optional //Question ids
surveydone int Optional //Did I finish the survey?
timecreated int Optional //Time of creation
timemodified int Optional //Time of last modification
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```



**XML-RPC (PHP structure)**

```
Array
(
 [surveys] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [template] => int
 [days] => int
 [questions] => string
 [surveydone] => int
 [timecreated] => int
 [timemodified] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="surveys">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="template">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="days">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="questions">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="surveydone">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

#### Error message

##### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

#### Restricted to logged-in users

Yes

#### Callable from AJAX

No



#### mod\_survey\_submit\_answers

Submit the answers for a given survey.

#### Arguments

**surveyid** (Required)

Survey id

##### General structure

```
int //Survey id
```

##### XML-RPC (PHP structure)

```
[surveyid] => int
```

**REST (POST parameters)**

```
surveyid= int
```

answers (Required)

**General structure**

```
list of (
object {
key string //Answer key
value string //Answer value
}
)
```

**XML-RPC (PHP structure)**

```
[answers] =>
Array
(
[0] =>
Array
(
[key] => string
[value] => string
)
)
```

**REST (POST parameters)**

```
answers[0][key]= string
answers[0][value]= string
```

**Response****General structure**

```
object {
status int //status: true if success
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

**XML-RPC (PHP structure)**

```
Array
(
[status] => int
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_survey\_view\_survey**

Trigger the course module viewed event and update the module completion status.

**Arguments****surveyid** (Required)

survey instance id

**General structure**

```
int //survey instance id
```

**XML-RPC (PHP structure)**

```
[surveyid] => int
```

**REST (POST parameters)**

```
surveyid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_url\_get\_urls\_by\_courses**

Returns a list of urls in a provided list of courses, if no list is provided all urls that the user can view will be returned.

**Arguments****courseids** (Default to "Array ()")

Array of course ids

**General structure**

```
Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)
```

**XML-RPC (PHP structure)**

```
[courseids] =>
 Array
 (
 [0] => int
)
```

**REST (POST parameters)**

courseids[0]= int

**Response**

**General structure**

```
object {
 urls list of (
 object {
 id int //Activity instance id
 coursemodule int //Course module id
 course int //Course id
 name string //Activity name
 intro string //Activity introduction
 introformat int //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 introfiles Optional //Files in the introduction
 list of (
 //File
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for external files.
 }
)section int Optional //Course section id
 visible int Optional //Visible
 groupmode int Optional //Group mode
 groupingid int Optional //Group id
 lang string Optional //Forced activity language
 externalurl string //External URL
 display int //How to display the url
 displayoptions string //Display options (width, height)
 parameters string //Parameters to append to the URL
 timemodified int //Last time the url was modified
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [
 [urls] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [externalurl] => string
 [display] => int
 [displayoptions] => string
 [parameters] => string
 [timemodified] => int
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="urls">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="externalurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="display">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displayoptions">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="parameters">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

`mod_url_view_url`



Trigger the course module viewed event and update the module completion status.

### Arguments

`urlid` (Required)

url instance id

#### General structure

```
int //url instance id
```

#### XML-RPC (PHP structure)

```
[urlid] => int
```

**REST (POST parameters)**

```
urlid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
})}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

[mod\\_wiki\\_edit\\_page](#)

Save the contents of a page.

**Arguments**

**pageid** (Required)

Page ID.

**General structure**

```
int //Page ID.
```

**XML-RPC (PHP structure)**

```
[pageid] => int
```

**REST (POST parameters)**

```
pageid= int
```

**content** (Required)

Page contents.

**General structure**

```
string //Page contents.
```

**XML-RPC (PHP structure)**

```
[content] => string
```

**REST (POST parameters)**

```
content= string
```

**section** (Default to "null")

Section page title.

**General structure**

```
string Default to "null" //Section page title.
```

**XML-RPC (PHP structure)**

```
[section] => string
```

**REST (POST parameters)**

```
section= string
```

**Response****General structure**

```
object {
 pageid int //Edited page id.
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [pageid] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

### Error message

#### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_wiki\_get\_page\_contents

Returns the contents of a page.

### Arguments

**pageid** (Required)

Page ID.

#### General structure

```
int //Page ID.
```

#### XML-RPC (PHP structure)

```
[pageid] => int
```

#### REST (POST parameters)

```
pageid= int
```

### Response

**General structure**

```
object {
page //Page
object {
id int //Page ID.
wikiid int //Page's wiki ID.
subwikiid int //Page's subwiki ID.
groupid int //Page's group ID.
userid int //Page's user ID.
title string //Page title.
cachedcontent string //Page contents.
contentformat int Optional //cachedcontent format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
caneditpage int //True if user can edit the page.
version int Optional //Latest version of the page.
tags Optional //Tags
list of (
object {
id int //Tag id.
name string //Tag name.
rawname string //The raw, unnormalised name for the tag as entered by users.
isstandard int //Whether this tag is standard.
tagcollid int //Tag collection id.
taginstanceid int //Tag instance id.
taginstancecontextid int //Context the tag instance belongs to.
itemid int //Id of the record tagged.
ordering int //Tag ordering.
flag int //Whether the tag is flagged as inappropriate.
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```



**XML-RPC (PHP structure)**

```
Array
(
 [page] =>
 Array
 (
 [id] => int
 [wikiid] => int
 [subwikiid] => int
 [groupid] => int
 [userid] => int
 [title] => string
 [cachedcontent] => string
 [contentformat] => int
 [caneditpage] => int
 [version] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="page">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="wikiid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="subwikiid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cachedcontent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="caneditpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="version">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</RESPONSE>
```

```

</KEY>
<KEY name="warningcode">
 <VALUE>string</VALUE>
</KEY>
<KEY name="message">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_wiki\_get\_page\_for\_editing

Locks and retrieves info of page-section to be edited.

#### Arguments

##### pageid (Required)

Page ID to edit.

#### General structure

```
int //Page ID to edit.
```



#### XML-RPC (PHP structure)

```
[pageid] => int
```

#### REST (POST parameters)

```
pageid= int
```

##### section (Default to "null")

Section page title.

#### General structure

```
string Default to "null" //Section page title.
```

**XML-RPC (PHP structure)**

```
[section] => string
```

**REST (POST parameters)**

```
section= string
```

**lockonly** (Default to "")

Just renew lock and not return content.

**General structure**

```
int Default to "" //Just renew lock and not return content.
```

**XML-RPC (PHP structure)**

```
[lockonly] => int
```

**REST (POST parameters)**

```
lockonly= int
```

**Response****General structure**

```
object {
 pagesection object {
 content string Optional //The contents of the page-section to be edited.
 contentformat string Optional //Format of the original content of the page.
 version int Optional //Latest version of the page.
 warnings Optional //List of warnings
 }
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [pagesection] =>
 Array
 (
 [content] => string
 [contentformat] => string
 [version] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pagesection">
 <SINGLE>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="version">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_wiki\_get\_subwiki\_files**

Returns the list of files for a specific subwiki.

**Arguments****wikiid** (Required)

Wiki instance ID.

**General structure**

```
int //Wiki instance ID.
```

**XML-RPC (PHP structure)**

```
[wikiid] => int
```

**REST (POST parameters)**

```
wikiid= int
```

**groupid (Default to "-1")**

Subwiki's group ID, -1 means current group. It will be ignored if the wiki doesn't use groups.

**General structure**

```
int Default to "-1" //Subwiki's group ID, -1 means current group. It will be ignored if the wiki doesn't use groups.
```

**XML-RPC (PHP structure)**

```
[groupid] => int
```

**REST (POST parameters)**

```
groupid= int
```

**userid (Default to "0")**

Subwiki's user ID, 0 means current user. It will be ignored in collaborative wikis.

**General structure**

```
int Default to "0" //Subwiki's user ID, 0 means current user. It will be ignored in collaborative wikis.
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response**

## General structure

```
object {
files //Files
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

## XML-RPC (PHP structure)

```

Array
(
[files] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)

```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

## mod\_wiki\_get\_subwiki\_pages

Returns the list of pages for a specific subwiki.

### Arguments

wikiid (Required)

Wiki instance ID.

#### General structure

```
int //Wiki instance ID.
```

#### XML-RPC (PHP structure)

```
[wikiid] => int
```

#### REST (POST parameters)

```
wikiid= int
```

groupid (Default to "-1")

Subwiki's group ID, -1 means current group. It will be ignored if the wiki doesn't use groups.

#### General structure

```
int Default to "-1" //Subwiki's group ID, -1 means current group. It will be ignored if the wiki doesn't use groups.
```

#### XML-RPC (PHP structure)

```
[groupid] => int
```

#### REST (POST parameters)

```
groupid= int
```

userid (Default to "0")

Subwiki's user ID, 0 means current user. It will be ignored in collaborative wikis.

#### General structure

```
int Default to "0" //Subwiki's user ID, 0 means current user. It will be ignored in collaborative wikis.
```

#### XML-RPC (PHP structure)

```
[userid] => int
```

#### REST (POST parameters)

```
userid= int
```

options (Default to "Array ()")

Options

**General structure**

```
Default to "Array
(
)
" //Options
object {
 sortby string Default to "title" //Field to sort by (id, title, ...).
 sortdirection string Default to "ASC" //Sort direction: ASC or DESC.
 includecontent int Default to "1" //Include each page contents or just the contents size.
}
```

**XML-RPC (PHP structure)**

```
[options] =>
 Array
 (
 [sortby] => string
 [sortdirection] => string
 [includecontent] => int
)
```

**REST (POST parameters)**

```
options[sortby]= string
options[sortdirection]= string
options[includecontent]= int
```

**Response**

**General structure**

```
object {
 pages list of (
 //Pages
 object {
 id int //Page ID.
 subwikiid int //Page's subwiki ID.
 title string //Page title.
 timecreated int //Time of creation.
 timemodified int //Time of last modification.
 timerendered int //Time of last renderization.
 userid int //ID of the user that last modified the page.
 pageviews int //Number of times the page has been viewed.
 readonly int //1 if readonly, 0 otherwise.
 caneditpage int //True if user can edit the page.
 firstpage int //True if it's the first page.
 cachedcontent string Optional //Page contents.
 contentformat int Optional //cachedcontent format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 contentsize int Optional //Size of page contents in bytes (doesn't include size of attached files).
 tags Optional //Tags
 list of (
 object {
 id int //Tag id.
 name string //Tag name.
 rawname string //The raw, unnormalised name for the tag as entered by users.
 isstandard int //Whether this tag is standard.
 tagcollid int //Tag collection id.
 taginstanceid int //Tag instance id.
 taginstancecontextid int //Context the tag instance belongs to.
 itemid int //Id of the record tagged.
 ordering int //Tag ordering.
 flag int //Whether the tag is flagged as inappropriate.
 }
)}
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [pages] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [subwikiid] => int
 [title] => string
 [timecreated] => int
 [timemodified] => int
 [timerendered] => int
 [userid] => int
 [pageviews] => int
 [readonly] => int
 [caneditpage] => int
 [firstpage] => int
 [cachedcontent] => string
 [contentformat] => int
 [contentsize] => int
 [tags] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [rawname] => string
 [isstandard] => int
 [tagcollid] => int
 [taginstanceid] => int
 [taginstancecontextid] => int
 [itemid] => int
 [ordering] => int
 [flag] => int
)
)
)
)
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pages">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="subwikiid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timerendered">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pageviews">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="readonly">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canneditpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="firstpage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cachedcontent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contentsize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tags">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rawname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isstandard">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tagcollid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstanceid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taginstancecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ordering">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="flag">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_wiki\_get\_subwikis



Returns the list of subwikis the user can see in a specific wiki.

### Arguments

`wikiid` (Required)

Wiki instance ID.

#### General structure

```
int //Wiki instance ID.
```

#### XML-RPC (PHP structure)

```
[wikiid] => int
```

## REST (POST parameters)

wikiid= int

## Response

## General structure

```

object {
 subwikis list of (
 //Subwikis
 object {
 id int //Subwiki ID.
 wikiid int //Wiki ID.
 groupid string //Group ID.
 userid int //User ID.
 canedit int //True if user can edit the subwiki.
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

## XML-RPC (PHP structure)

```

Array
(
 [
 [subwikis] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [wikiid] => int
 [groupid] => string
 [userid] => int
 [canedit] => int
)
)
)
 [
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="subwikis">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="wikiid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cannedit">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_wiki\_get\_wikis\_by\_courses**

Returns a list of wiki instances in a provided set of courses, if no courses are provided then all the wiki instances the user has access to will be returned.

### Arguments

courseids (Default to "Array ()")

Array of course ids.

#### General structure

```
Default to "Array
(
)
" //Array of course ids.
list of (
int //Course ID
)
```

#### XML-RPC (PHP structure)

```
[courseids] =>
 Array
 (
 [0] => int
)
```

#### REST (POST parameters)

```
courseids[0]= int
```

### Response



**General structure**

```
object {
wikis list of (
 //Wikis
object {
id int //Activity instance id
coursemodule int //Course module id
course int //Course id
name string //Activity name
intro string Optional //Activity introduction
introformat int Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
introfiles Optional //Files in the introduction
list of (
 //File
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)section int Optional //Course section id
visible int Optional //Visible
groupmode int Optional //Group mode
groupingid int Optional //Group id
lang string Optional //Forced activity language
timecreated int Optional //Time of creation.
timemodified int Optional //Time of last modification.
firstpagetitle string Optional //First page title.
wikimode string Optional //Wiki mode (individual, collaborative).
defaultformat string Optional //Wiki's default format (html, creole, nwiki).
forceformat int Optional //1 if format is forced, 0 otherwise.
editbegin int Optional //Edit begin.
editend int Optional //Edit end.
cancreatepages int //True if user can create pages.
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```



**XML-RPC (PHP structure)**

```
Array
(
 [wikis] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [coursemodule] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [introfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [section] => int
 [visible] => int
 [groupmode] => int
 [groupingid] => int
 [lang] => string
 [timecreated] => int
 [timemodified] => int
 [firstpagetitle] => string
 [wikimode] => string
 [defaultformat] => string
 [forceformat] => int
 [editbegin] => int
 [editend] => int
 [cancreatepages] => int
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="wikis">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursemodule">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="section">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupmode">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="groupingid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="firstpagetitle">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="wikimode">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="defaultformat">
 <VALUE>string</VALUE>
</KEY>
<KEY name="forceformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="editbegin">
 <VALUE>int</VALUE>
</KEY>
<KEY name="editend">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cancreatepages">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```



### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_wiki\_new\_page

Create a new page in a subwiki.

### Arguments

**title** (Required)

New page title.

**General structure**

```
string //New page title.
```

**XML-RPC (PHP structure)**

```
[title] => string
```

**REST (POST parameters)**

```
title= string
```

**content** (Required)

Page contents.

**General structure**

```
string //Page contents.
```

**XML-RPC (PHP structure)**

```
[content] => string
```

**REST (POST parameters)**

```
content= string
```

**contentformat** (Default to "null")

Page contents format. If an invalid format is provided, default wiki format is used.

**General structure**

```
string Default to "null" //Page contents format. If an invalid format is provided, default
wiki format is used.
```

**XML-RPC (PHP structure)**

```
[contentformat] => string
```

**REST (POST parameters)**

```
contentformat= string
```

**subwikiid** (Default to "null")

Page's subwiki ID.

**General structure**

```
int Default to "null" //Page's subwiki ID.
```

**XML-RPC (PHP structure)**

```
[subwikiid] => int
```

**REST (POST parameters)**

```
subwikiid= int
```

**wikiid** (Default to "null")

Page's wiki ID. Used if subwiki does not exists.

**General structure**

```
int Default to "null" //Page's wiki ID. Used if subwiki does not exists.
```

**XML-RPC (PHP structure)**

```
[wikiid] => int
```

**REST (POST parameters)**

```
wikiid= int
```

**userid (Default to "null")**

Subwiki's user ID. Used if subwiki does not exists.

**General structure**

```
int Default to "null" //Subwiki's user ID. Used if subwiki does not exists.
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**groupid (Default to "null")**

Subwiki's group ID. Used if subwiki does not exists.

**General structure**

```
int Default to "null" //Subwiki's group ID. Used if subwiki does not exists.
```

**XML-RPC (PHP structure)**

```
[groupid] => int
```

**REST (POST parameters)**

```
groupid= int
```

**Response****General structure**

```
object {
 pageid int //New page id.
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [pageid] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
}
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="pageid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

No

mod\_wiki\_view\_page

Trigger the page viewed event and update the module completion status.

## Arguments

pageid (Required)

Wiki page ID.

### General structure

```
int //Wiki page ID.
```

### XML-RPC (PHP structure)

```
[pageid] => int
```

### REST (POST parameters)

```
pageid= int
```

## Response

### General structure

```
object {
 status int //Status: true if success.
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

### XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_wiki\_view\_wiki**

Trigger the course module viewed event and update the module completion status.

**Arguments****wikiid** (Required)

Wiki instance ID.

**General structure**

```
int //Wiki instance ID.
```

**XML-RPC (PHP structure)**

```
[wikiid] => int
```

## REST (POST parameters)

```
wikiid= int
```

## Response

## General structure

```
object {
 status int //Status: true if success.
 warnings Optional //List of warnings
 list of (
 //warning
)
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```

## XML-RPC (PHP structure)

```
Array
(
 [
 [status] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
]
)
]
)
]
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

[mod\\_workshop\\_add\\_submission](#)

Add a new submission to a given workshop.

**Arguments**

**workshopid** (Required)

Workshop id

**General structure**

```
int //Workshop id
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**title** (Required)

Submission title

**General structure**

```
string //Submission title
```

**XML-RPC (PHP structure)**

```
[title] => string
```

**REST (POST parameters)**

```
title= string
```

**content** (Default to "")

Submission text content

**General structure**

```
string Default to "" //Submission text content
```

**XML-RPC (PHP structure)**

```
[content] => string
```

**REST (POST parameters)**

```
content= string
```

**contentformat** (Default to "o")

The format used for the content

**General structure**

```
int Default to "0" //The format used for the content
```

**XML-RPC (PHP structure)**

```
[contentformat] => int
```

**REST (POST parameters)**

```
contentformat= int
```

**inlineattachmentsid** (Default to "o")

The draft file area id for inline attachments in the content

**General structure**

```
int Default to "0" //The draft file area id for inline attachments in the content
```

**XML-RPC (PHP structure)**

```
[inlineattachmentsid] => int
```

**REST (POST parameters)**

```
inlineattachmentsid= int
```

**attachmentsid** (Default to "o")

The draft file area id for attachments

**General structure**

```
int Default to "0" //The draft file area id for attachments
```

**XML-RPC (PHP structure)**

```
[attachmentsid] => int
```

**REST (POST parameters)**

```
attachmentsid= int
```

**Response**

## General structure

```
object {
status int //True if the submission was created false otherwise.
submissionid int Optional //New workshop submission id.
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})
```

## XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [submissionid] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_delete\_submission**

Deletes the given submission.

**Arguments****submissionid** (Required)

Submission id

**General structure**

```
int //Submission id
```

**XML-RPC (PHP structure)**

```
[submissionid] => int
```

**REST (POST parameters)**

```
submissionid= int
```

**Response****General structure**

```
object {
 status int //True if the submission was deleted.
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

No

#### mod\_workshop\_evaluate\_assessment

Evaluates an assessment (used by teachers for provide feedback to the reviewer).

## Arguments

**assessmentid** (Required)

Assessment id.

### General structure

```
int //Assessment id.
```

### XML-RPC (PHP structure)

```
[assessmentid] => int
```

### REST (POST parameters)

```
assessmentid= int
```

**feedbacktext** (Default to "")

The feedback for the reviewer.

### General structure

```
string Default to "" //The feedback for the reviewer.
```

### XML-RPC (PHP structure)

```
[feedbacktext] => string
```

### REST (POST parameters)

```
feedbacktext= string
```

**feedbackformat** (Default to "0")

The feedback format for text.

### General structure

```
int Default to "0" //The feedback format for text.
```

### XML-RPC (PHP structure)

```
[feedbackformat] => int
```



### REST (POST parameters)

```
feedbackformat= int
```

**weight** (Default to "1")

The new weight for the assessment.

### General structure

```
int Default to "1" //The new weight for the assessment.
```

### XML-RPC (PHP structure)

```
[weight] => int
```

**REST (POST parameters)**

```
weight= int
```

**gradinggradeover** (Default to "")

The new grading grade.

**General structure**

```
string Default to "" //The new grading grade.
```

**XML-RPC (PHP structure)**

```
[gradinggradeover] => string
```

**REST (POST parameters)**

```
gradinggradeover= string
```

**Response****General structure**

```
object {
 status int //status: true if the assessment was evaluated, false otherwise.
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_evaluate\_submission**

Evaluates a submission (used by teachers for provide feedback or override the submission grade).

**Arguments****submissionid** (Required)

submission id.

**General structure**

```
int //submission id.
```

**XML-RPC (PHP structure)**

```
[submissionid] => int
```

**REST (POST parameters)**

```
submissionid= int
```

**feedbacktext (Default to "")**

The feedback for the author.

**General structure**

```
string Default to "" //The feedback for the author.
```

**XML-RPC (PHP structure)**

```
[feedbacktext] => string
```

**REST (POST parameters)**

```
feedbacktext= string
```

**feedbackformat (Default to "o")**

The feedback format for text.

**General structure**

```
int Default to "0" //The feedback format for text.
```

**XML-RPC (PHP structure)**

```
[feedbackformat] => int
```

**REST (POST parameters)**

```
feedbackformat= int
```

**published (Default to "")**

Publish the submission for others?

**General structure**

```
int Default to "" //Publish the submission for others?.
```

**XML-RPC (PHP structure)**

```
[published] => int
```

**REST (POST parameters)**

```
published= int
```

**gradeover (Default to "")**

The new submission grade.

**General structure**

```
string Default to "" //The new submission grade.
```

**XML-RPC (PHP structure)**

```
[gradeover] => string
```

**REST (POST parameters)**

```
gradeover= string
```

**Response****General structure**

```
object {
 status int //status: true if the submission was evaluated, false otherwise.
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_get\_assessment**

Retrieves the given assessment.

**Arguments****assessmentid** (Required)

Assessment id

**General structure**

```
int //Assessment id
```

**XML-RPC (PHP structure)**

```
[assessmentid] => int
```

**REST (POST parameters)**

```
assessmentid= int
```

**Response**

**General structure**

```

object {
assessment object {
id int //The primary key of the record.
submissionid int //The id of the assessed submission
reviewerid int //The id of the reviewer who makes this assessment
weight int //The weight of the assessment for the purposes of aggregation
timecreated int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 If greater than 0 then the timestamp of when the reviewer assessed for the first time
timemodified int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 If greater than 0 then the timestamp of when the reviewer assessed for the last time
grade double //The aggregated grade for submission suggested by the reviewer.
 The grade 0..100 is computed from the values assigned to the assessment dimensions fields. If NULL then it has not been aggregated yet.
gradinggrade double //The computed grade 0..100 for this assessment. If NULL then it has not been computed yet.
gradinggradeover double //Grade for the assessment manually overridden by a teacher.
 Grade is always from interval 0..100. If NULL then the grade is not overriden.
gradinggradeoverby int //The id of the user who has overridden the grade for submission.
feedbackauthor string //The comment/feedback from the reviewer for the author.
feedbackauthorformat int Default to "1" //feedbackauthor format (1 = HTML, 0 = MOODLE, or 4 = MARKDOWN
feedbackauthorattachment int //Are there some files attached to the feedbackauthor field?
 Sets to 1 by file_postupdate_standard_filemanager().
feedbackreviewer string Optional //The comment/feedback from the teacher for the reviewer.
 For example the reason why the grade for assessment was overridden
feedbackreviewerformat int Default to "1" //feedbackreviewer format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
feedbackcontentfiles //feedbackcontentfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
)feedbackattachmentfiles //feedbackattachmentfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
})
))
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)
}

```



**XML-RPC (PHP structure)**

```
Array
(
 [assessment] =>
 Array
 (
 [id] => int
 [submissionid] => int
 [reviewerid] => int
 [weight] => int
 [timecreated] => int
 [timemodified] => int
 [grade] => double
 [gradinggrade] => double
 [gradinggradeover] => double
 [gradinggradeoverby] => int
 [feedbackauthor] => string
 [feedbackauthorformat] => int
 [feedbackauthorattachment] => int
 [feedbackreviewer] => string
 [feedbackreviewerformat] => int
 [feedbackcontentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [feedbackattachmentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="assessment">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="weight">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthor">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackauthorformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthorattachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackreviewer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackreviewerformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackcontentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



#### Error message

##### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

#### Restricted to logged-in users

Yes

#### Callable from AJAX

No

[mod\\_workshop\\_get\\_assessment\\_form\\_definition](#)

Retrieves the assessment form definition.

#### Arguments

**assessmentid** (Required)

Assessment id

##### General structure

```
int //Assessment id
```

##### XML-RPC (PHP structure)

```
[assessmentid] => int
```

##### REST (POST parameters)

```
assessmentid= int
```

**mode** (Default to "assessment")

The form mode (assessment or preview)

##### General structure

```
string Default to "assessment" //The form mode (assessment or preview)
```

##### XML-RPC (PHP structure)

```
[mode] => string
```

##### REST (POST parameters)

```
mode= string
```

#### Response



**General structure**

```
object {
dimensionscount int //The number of dimensions used by the form.
descriptionfiles //Files in the description text
list of (
 //File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
}
)options //The form options.
list of (
object {
name string //Option name.
value string //Option value.
}
)fields //The form fields.
list of (
object {
name string //Field name.
value string //Field default value.
}
)current //The current field values.
list of (
object {
name string //Field name.
value string //Current field value.
}
)dimensionsinfo //The dimensions general information.
list of (
object {
id int //Dimension id.
min double //Minimum grade for the dimension.
max double //Maximum grade for the dimension.
weight string //The weight of the dimension.
scale string Optional //Scale items (if used).
}
)warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
})}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [dimensionscount] => int
 [descriptionfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [options] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [value] => string
)
)
)
 [fields] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [value] => string
)
)
)
 [current] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [value] => string
)
)
)
 [dimensionsinfo] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [min] => double
 [max] => double
 [weight] => string
 [scale] => string
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="dimensionscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="descriptionfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="fields">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="current">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="dimensionsinfo">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="min">
 <VALUE>double</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </RESPONSE>
```



```

<KEY name="max">
 <VALUE>double</VALUE>
</KEY>
<KEY name="weight">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scale">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes



### Callable from AJAX

No

### mod\_workshop\_get\_grades

Returns the assessment and submission grade for the given user.

### Arguments

**workshopid** (Required)

Workshop instance id.

#### General structure

```
int //Workshop instance id.
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**userid** (Default to "0")

User id (empty or 0 for current user).

**General structure**

```
int Default to "0" //User id (empty or 0 for current user).
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response****General structure**

```
object {
 assessmentrawgrade double Optional //The assessment raw (numeric) grade.
 assessmentlongstrgrade string Optional //The assessment string grade.
 assessmentgradehidden int Optional //Whether the grade is hidden or not.
 submissionrawgrade double Optional //The submission raw (numeric) grade.
 submissionlongstrgrade string Optional //The submission string grade.
 submissiongradehidden int Optional //Whether the grade is hidden or not.
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [assessmentrawgrade] => double
 [assessmentlongstrgrade] => string
 [assessmentgradehidden] => int
 [submissionrawgrade] => double
 [submissionlongstrgrade] => string
 [submissiongradehidden] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="assessmentrawgrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="assessmentlongstrgrade">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="assessmentgradehidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionrawgrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="submissionlongstrgrade">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="submissiongradehidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_get\_grades\_report**

Retrieves the assessment grades report.

**Arguments**

workshopid (Required)

Workshop instance id.

#### General structure

```
int //Workshop instance id.
```

#### XML-RPC (PHP structure)

```
[workshopid] => int
```

#### REST (POST parameters)

```
workshopid= int
```

**groupid** (Default to "o")

Group id, o means that the function will determine the user group.

#### General structure

```
int Default to "0" //Group id, 0 means that the function will determine the user group.
```

#### XML-RPC (PHP structure)

```
[groupid] => int
```

#### REST (POST parameters)

```
groupid= int
```

**sortby** (Default to "lastname")

sort by this element: lastname, firstname, submissiontitle, submissionmodified, submissiongrade, gradinggrade.

#### General structure

```
string Default to "lastname" //sort by this element: Lastname, firstname, submissiontitle,
submissionmodified, submissiongrade, gradinggrade.
```

#### XML-RPC (PHP structure)

```
[sortby] => string
```

#### REST (POST parameters)

```
sortby= string
```

**sortdirection** (Default to "ASC")

sort direction: ASC or DESC

#### General structure

```
string Default to "ASC" //sort direction: ASC or DESC
```

#### XML-RPC (PHP structure)

```
[sortdirection] => string
```

#### REST (POST parameters)

```
sortdirection= string
```

**page** (Default to "o")

The page of records to return.

**General structure**

```
int Default to "0" //The page of records to return.
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

**perpage** (Default to "0")

The number of records to return per page.

**General structure**

```
int Default to "0" //The number of records to return per page.
```

**XML-RPC (PHP structure)**

```
[perpage] => int
```

**REST (POST parameters)**

```
perpage= int
```

**Response**



**General structure**

```
object {
report object {
grades list of (
object {
userid int //The id of the user being displayed in the report.
submissionid int //Submission id.
submissiontitle string //Submission title.
submissionmodified int //Timestamp submission was updated.
submissiongrade double Optional //Aggregated grade for the submission.
gradinggrade double Optional //Computed grade for the assessment.
submissiongradeover double Optional //Grade for the assessment overrided
by the teacher.
submissiongradeoverby int Optional //The id of the user who overrided
the grade.
submissionpublished int Optional //Whether is a submission published.
reviewedby Optional //The users who reviewed the
user submission.

list of (
object {
userid int //The id of the user (0 when is configured to do not display names).
assessmentid int //The id of the assessment.
submissionid int //The id of the submission assessed.
grade double //The grade for submission.
gradinggrade double //The grade for assessment.
gradinggradeover double //The aggregated grade overrided.
weight int //The weight of the assessment for aggregation.
}
)reviewerof Optional //The assessments the user
reviewed.

list of (
object {
userid int //The id of the user (0 when is configured to do not display names).
assessmentid int //The id of the assessment.
submissionid int //The id of the submission assessed.
grade double //The grade for submission.
gradinggrade double //The grade for assessment.
gradinggradeover double //The aggregated grade overrided.
weight int //The weight of the assessment for aggregation.
}
)
)}
)totalcount int //Number of total submissions.
}

warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
)}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [report] =>
 Array
 (
 [
 [grades] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [submissionid] => int
 [submissiontitle] => string
 [submissionmodified] => int
 [submissiongrade] => double
 [gradinggrade] => double
 [submissiongradeover] => double
 [submissiongradeoverby] => int
 [submissionpublished] => int
 [reviewedby] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [assessmentid] => int
 [submissionid] => int
 [grade] => double
 [gradinggrade] => double
 [gradinggradeover] => double
 [weight] => int
)
)
)
)
 [reviewerof] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [assessmentid] => int
 [submissionid] => int
 [grade] => double
 [gradinggrade] => double
 [gradinggradeover] => double
 [weight] => int
)
)
)
)
)
)
)
 [totalcount] => int
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="report">
 <SINGLE>
 <KEY name="grades">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissiontitle">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="submissionmodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissiongrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="submissiongradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="submissiongradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionpublished">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewedby">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="assessmentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="weight">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="reviewerof">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="assessmentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeover">
```



```

<VALUE>double</VALUE>
</KEY>
<KEY name="weight">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="totalcount">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### `mod_workshop_get_reviewer_assessments`

Retrieves all the assessments reviewed by the given user.

### Arguments

`workshopid` (Required)

Workshop instance id.

**General structure**

```
int //Workshop instance id.
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**userid (Default to "0")**

User id who did the assessment review (empty or 0 for current user).

**General structure**

```
int Default to "0" //User id who did the assessment review (empty or 0 for current user).
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response**

**General structure**

```

object {
 assessments list of (
 object {
 id int //The primary key of the record.
 submissionid int //The id of the assessed submission
 reviewerid int //The id of the reviewer who makes this assessment
 weight int //The weight of the assessment for the purposes of aggregation
 timecreated int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 If greater than 0 then the timestamp of when the reviewer assessed for the first time
 timemodified int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 If greater than 0 then the timestamp of when the reviewer assessed for the last time
 grade double //The aggregated grade for submission suggested by the reviewer.
 The grade 0..100 is computed from the values assigned to the assessment dimensions fields. If NULL then it has not been aggregated yet.
 gradinggrade double //The computed grade 0..100 for this assessment. If NULL then it has not been computed yet.
 gradinggradeover double //Grade for the assessment manually overridden by a teacher.
 Grade is always from interval 0..100. If NULL then the grade is not overridden.
 gradinggradeoverby int //The id of the user who has overridden the grade for submission.
 feedbackauthor string //The comment/feedback from the reviewer for the author.
 feedbackauthorformat int Default to "1" //feedbackauthor format (1 = HTML, 0 = MOODLE, or 4 = MARKDOWN
 feedbackauthorattachment int //Are there some files attached to the feedbackauthor field?
 Sets to 1 by file_postupdate_standard_filemanager().
 feedbackreviewer string Optional //The comment/feedback from the teacher for the reviewer.
 For example the reason why the grade for assessment was overridden
 feedbackreviewerformat int Default to "1" //feedbackreviewer format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 feedbackcontentfiles //feedbackcontentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)feedbackattachmentfiles //feedbackattachmentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
)
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [assessments] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [submissionid] => int
 [reviewerid] => int
 [weight] => int
 [timecreated] => int
 [timemodified] => int
 [grade] => double
 [gradinggrade] => double
 [gradinggradeover] => double
 [gradinggradeoverby] => int
 [feedbackauthor] => string
 [feedbackauthorformat] => int
 [feedbackauthorattachment] => int
 [feedbackreviewer] => string
 [feedbackreviewerformat] => int
 [feedbackcontentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [feedbackattachmentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="assessments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="weight">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthor">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackauthorformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthorattachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackreviewer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackreviewerformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackcontentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

```

<KEY name="feedbackattachmentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

mod\_workshop\_get\_submission

Retrieves the given submission.

**Arguments**

submissionid (Required)

Submission id

**General structure**

```
int //Submission id
```

**XML-RPC (PHP structure)**

```
[submissionid] => int
```

**REST (POST parameters)**

```
submissionid= int
```

**Response**

**General structure**

```

object {
 submission object {
 id int //The primary key of the record.
 workshopid int //The id of the workshop instance.
 example int //Is this submission an example from teacher.
 authorid int //The author of the submission.
 timecreated int //Timestamp when the work was submitted for the first time.
 timemodified int //Timestamp when the submission has been updated.
 title string //The submission title.
 content string //Submission text.
 contentformat int Default to "1" //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 contenttrust int //The trust mode of the data.
 attachment int //Used by File API file_postupdate_standard_filemanager.
 grade double Optional //Aggregated grade for the submission. The grade is a decimal number from interval 0..100.
 If NULL then the grade for submission has not been aggregated yet.
 gradeover double Optional //Grade for the submission manually overridden by a teacher. Grade is always from interval 0..100.
 If NULL then the grade is not overridden.
 gradeoverby int Optional //The id of the user who has overridden the grade for submission.
 feedbackauthor string Optional //Teacher comment/feedback for the author of the submission, for example describing the reasons
 for the grade overriding.
 feedbackauthorformat int Default to "1" //feedbackauthor format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 timegraded int Optional //The timestamp when grade or gradeover was recently modified.
 published int //Shall the submission be available to other when the workshop is closed.
 late int //Has this submission been submitted after the deadline or during the assessment phase?
 contentfiles Optional //contentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)attachmentfiles Optional //attachmentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)
})
warnings Optional //List of warnings
list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [submission] =>
 Array
 (
 [id] => int
 [workshopid] => int
 [example] => int
 [authorid] => int
 [timecreated] => int
 [timemodified] => int
 [title] => string
 [content] => string
 [contentformat] => int
 [contenttrust] => int
 [attachment] => int
 [grade] => double
 [gradeover] => double
 [gradeoverby] => int
 [feedbackauthor] => string
 [feedbackauthorformat] => int
 [timegraded] => int
 [published] => int
 [late] => int
 [contentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [attachmentfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="submission">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="workshopid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="example">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="authorid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contenttrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthor">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackauthorformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timegraded">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="published">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="late">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
```



```

 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="attachmentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

No

### mod\_workshop\_get\_submission\_assessments

Retrieves all the assessments of the given submission.

#### Arguments

submissionid (Required)

Submission id

##### General structure

```
int //Submission id
```

##### XML-RPC (PHP structure)

```
[submissionid] => int
```

##### REST (POST parameters)

```
submissionid= int
```

#### Response



**General structure**

```

object {
 assessments list of (
 object {
 id int //The primary key of the record.
 submissionid int //The id of the assessed submission
 reviewerid int //The id of the reviewer who makes this assessment
 weight int //The weight of the assessment for the purposes of aggregation
 timecreated int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 //If greater than 0 then the timestamp of when the reviewer assessed for the first time
 timemodified int //If 0 then the assessment was allocated but the reviewer has not assessed yet.
 //If greater than 0 then the timestamp of when the reviewer assessed for the last time
 grade double //The aggregated grade for submission suggested by the reviewer.
 //The grade 0..100 is computed from the values assigned to the assessment dimensions fields. If NULL then it has not been aggregated yet.
 gradinggrade double //The computed grade 0..100 for this assessment. If NULL then it has not been computed yet.
 gradinggradeover double //Grade for the assessment manually overridden by a teacher.
 //Grade is always from interval 0..100. If NULL then the grade is not overridden.
 gradinggradeoverby int //The id of the user who has overridden the grade for submission.
 feedbackauthor string //The comment/feedback from the reviewer for the author.
 feedbackauthorformat int Default to "1" //feedbackauthor format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 feedbackauthorattachment int //Are there some files attached to the feedbackauthor field?
 Sets to 1 by file_postupdate_standard_filemanager().
 feedbackreviewer string Optional //The comment/feedback from the teacher for the reviewer.
 For example the reason why the grade for assessment was overridden
 feedbackreviewerformat int Default to "1" //feedbackreviewer format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 feedbackcontentfiles //feedbackcontentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)feedbackattachmentfiles //feedbackattachmentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)
 })
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```



**XML-RPC (PHP structure)**

```
Array
(
 [
 [assessments] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [submissionid] => int
 [reviewerid] => int
 [weight] => int
 [timecreated] => int
 [timemodified] => int
 [grade] => double
 [gradinggrade] => double
 [gradinggradeover] => double
 [gradinggradeoverby] => int
 [feedbackauthor] => string
 [feedbackauthorformat] => int
 [feedbackauthorattachment] => int
 [feedbackreviewer] => string
 [feedbackreviewerformat] => int
 [feedbackcontentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [feedbackattachmentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
```



## REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="assessments">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="weight">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthor">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackauthorformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthorattachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackreviewer">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackreviewerformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackcontentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

```

<KEY name="feedbackattachmentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

**mod\_workshop\_get\_submissions**

Retrieves all the workshop submissions or the one done by the given user (except example submissions).

**Arguments****workshopid** (Required)

Workshop instance id.

**General structure**

```
int //Workshop instance id.
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**userid** (Default to "0")

Get submissions done by this user. Use 0 or empty for the current user

**General structure**

```
int Default to "0" //Get submissions done by this user. Use 0 or empty for the current user
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**groupid** (Default to "0")

Group id, 0 means that the function will determine the user group. It will return submissions done by users in the given group.

**General structure**

```
int Default to "0" //Group id, 0 means that the function will determine the user group.
It will return submissions done by users in the given group.
```

**XML-RPC (PHP structure)**

```
[groupid] => int
```

**REST (POST parameters)**

```
groupid= int
```

**page** (Default to "0")

The page of records to return.

**General structure**

```
int Default to "0" //The page of records to return.
```

**XML-RPC (PHP structure)**

```
[page] => int
```

**REST (POST parameters)**

```
page= int
```

**perpage** (Default to "0")

The number of records to return per page.

**General structure**

```
int Default to "0" //The number of records to return per page.
```

**XML-RPC (PHP structure)**

```
[perpage] => int
```

**REST (POST parameters)**

```
perpage= int
```

**Response**

**General structure**

```

object {
 submissions list of (
 object {
 id int //The primary key of the record.
 workshopid int //The id of the workshop instance.
 example int //Is this submission an example from teacher.
 authorid int //The author of the submission.
 timecreated int //Timestamp when the work was submitted for the first time.
 timemodified int //Timestamp when the submission has been updated.
 title string //The submission title.
 content string //Submission text.
 contentformat int Default to "1" //content format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 contenttrust int //The trust mode of the data.
 attachment int //Used by File API file_postupdate_standard_filemanager.
 grade double Optional //Aggregated grade for the submission. The grade is a decimal number from interval 0..100.
 If NULL then the grade for submission has not been aggregated yet.
 gradeover double Optional //Grade for the submission manually overridden by a teacher. Grade is always from interval 0..100.
 If NULL then the grade is not overriden.
 gradeoverby int Optional //The id of the user who has overridden the grade for submission.
 feedbackauthor string Optional //Teacher comment/feedback for the author of the submission, for example describing the reasons
 for the grade overriding.
 feedbackauthorformat int Default to "1" //feedbackauthor format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 timegraded int Optional //The timestamp when grade or gradeover was recently modified.
 published int //Shall the submission be available to other when the workshop is closed.
 late int //Has this submission been submitted after the deadline or during the assessment phase?
 contentfiles Optional //contentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)attachmentfiles Optional //attachmentfiles
 list of (
 object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
 }
)
 })
 totalcount int //Total count of submissions.
 totalfilesize int //Total size (bytes) of the files attached to all the
 submissions (even the ones not returned due to pagination).
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [submissions] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [workshopid] => int
 [example] => int
 [authorid] => int
 [timecreated] => int
 [timemodified] => int
 [title] => string
 [content] => string
 [contentformat] => int
 [contenttrust] => int
 [attachment] => int
 [grade] => double
 [gradeover] => double
 [gradeoverby] => int
 [feedbackauthor] => string
 [feedbackauthorformat] => int
 [timegraded] => int
 [published] => int
 [late] => int
 [contentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [attachmentfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
)
)
)
)
)
 [totalcount] => int
 [totalfilesize] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="submissions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="workshopid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="example">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="authorid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contenttrust">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="attachment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradeover">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradeoverby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="feedbackauthor">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="feedbackauthorformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timegraded">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="published">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="late">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contentfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="attachmentfiles">
<MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="totalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="totalfilesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
<MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</RESPONSE>

```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_get\_user\_plan**

Return the planner information for the given user.

**Arguments****workshopid** (Required)

Workshop instance id.

**General structure**

```
int //Workshop instance id.
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**userid** (Default to "0")

User id (empty or 0 for current user).

**General structure**

```
int Default to "0" //User id (empty or 0 for current user).
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response**

**General structure**

```
object {
userplan object {
phases list of (
object {
code int //Phase code.
title string //Phase title.
active int //Whether is the active task.
tasks list of (
object {
code string //Task code.
title string //Task title.
link string //Link to task.
details string Optional //Task details.
completed string //Completion information (maybe empty, maybe a boolean or generic info.
})
actions list of (
object {
type string Optional //Action type.
label string Optional //Action label.
url string //Link to action.
method string Optional //Get or post.
})
})
examples list of (
object {
id int //Example submission id.
title string //Example submission title.
assessmentid int //Example submission assessment id.
grade double //The submission grade.
gradinggrade double //The assessment grade.
})
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [userplan] =>
 Array
 (
 [
 [phases] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [code] => int
 [title] => string
 [active] => int
 [tasks] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [code] => string
 [title] => string
 [link] => string
 [details] => string
 [completed] => string
)
)
)
)
 [actions] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [type] => string
 [label] => string
 [url] => string
 [method] => string
)
)
)
)
)
)
)
)
)
 [examples] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [title] => string
 [assessmentid] => int
 [grade] => double
 [gradinggrade] => double
)
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="userplan">
 <SINGLE>
 <KEY name="phases">
 <MULTIPLE>
 <SINGLE>
 <KEY name="code">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="active">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="tasks">
 <MULTIPLE>
 <SINGLE>
 <KEY name="code">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="link">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="details">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="completed">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="actions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="label">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="method">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="examples">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="assessmentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
```



```

</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_workshop\_get\_workshop\_access\_information

Return access information for a given workshop.



### Arguments

workshopid (Required)

Workshop instance id.

#### General structure

```
int //Workshop instance id.
```

#### XML-RPC (PHP structure)

```
[workshopid] => int
```

#### REST (POST parameters)

```
workshopid= int
```

### Response

**General structure**

```
object {
 creatingsubmissionallowed int //Is the given user allowed to create their submission?
 modifyingsubmissionallowed int //Is the user allowed to modify his existing submission?
 assessingallowed int //Is the user allowed to create/edit his assessments?
 assessingexamplesallowed int //Are reviewers allowed to create/edit their assessments of the example submissions?
 examplesassessedbeforesubmission int //Whether the given user has assessed all his required examples before submission
 (always true if there are not examples to assess or not configured to check before submission).
 examplesassessedbeforeassessment int //Whether the given user has assessed all his required examples before assessment
 (always true if there are not examples to assessor not configured to check before assessment).
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)canview int //Whether the user has the capability mod/workshop:view allowed.
canaddinstance int //Whether the user has the capability mod/workshop:addinstance allowed.
canswitchphase int //Whether the user has the capability mod/workshop:switchphase allowed.
caneditdimensions int //Whether the user has the capability mod/workshop:editdimensions allowed.
cansubmit int //Whether the user has the capability mod/workshop:submit allowed.
canpeerassess int //Whether the user has the capability mod/workshop:peerassess allowed.
canmanageexamples int //Whether the user has the capability mod/workshop:manageexamples allowed.
canallocate int //Whether the user has the capability mod/workshop:allocate allowed.
canpublishsubmissions int //Whether the user has the capability mod/workshop:publishsubmissions allowed.
canviewauthornames int //Whether the user has the capability mod/workshop:viewauthornames allowed.
canviewreviewernames int //Whether the user has the capability mod/workshop:viewreviewernames allowed.
canviewallsubmissions int //Whether the user has the capability mod/workshop:viewallsubmissions allowed.
canviewpublishedsubmissions int //Whether the user has the capability mod/workshop:viewpublishedsubmissions allowed.
canviewauthorpublished int //Whether the user has the capability mod/workshop:viewauthorpublished allowed.
canviewallassessments int //Whether the user has the capability mod/workshop:viewallassessments allowed.
canoverridegrades int //Whether the user has the capability mod/workshop:overridegrades allowed.
canignoredeadlines int //Whether the user has the capability mod/workshop:ignoredeadlines allowed.
candeletesubmissions int //Whether the user has the capability mod/workshop:deletesubmissions allowed.
canexportsubmissions int //Whether the user has the capability mod/workshop:exportsubmissions allowed.
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [creatingsubmissionallowed] => int
 [modifyingsubmissionallowed] => int
 [assessingallowed] => int
 [assessingexamplesallowed] => int
 [examplesassessedbeforesubmission] => int
 [examplesassessedbeforeassessment] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
 [canview] => int
 [canaddinstance] => int
 [canswitchphase] => int
 [caneditdimensions] => int
 [cansubmit] => int
 [canpeerassess] => int
 [canmanageexamples] => int
 [canallocate] => int
 [canpublishsubmissions] => int
 [canviewauthornames] => int
 [canviewreviewernames] => int
 [canviewallsubmissions] => int
 [canviewpublishedsubmissions] => int
 [canviewauthorpublished] => int
 [canviewallassessments] => int
 [canoverridegrades] => int
 [canignoredeadlines] => int
 [candeletesubmissions] => int
 [canexportsubmissions] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="creatingsubmissionallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="modifyingsubmissionallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="assessingallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="assessingexamplesallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="examplesassessedbeforesubmission">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="examplesassessedbeforeassessment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="canview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canaddinstance">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canswitchphase">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="caneditdimensions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cansubmit">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canpeerassess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanageexamples">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canallocate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canpublishsubmissions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewauthornames">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewreviewernames">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewallsubmissions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewpublishedsubmissions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canviewauthorpublished">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>

```



```

<KEY name="canviewallassessments">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canoverridegrades">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canignoredeadlines">
 <VALUE>int</VALUE>
</KEY>
<KEY name="candeletesubmissions">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canexportsubmissions">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_workshop\_get\_workshops\_by\_courses

Returns a list of workshops in a provided list of courses, if no list is provided all workshops that the user can view will be returned.

### Arguments

courseids (Default to "Array ()")

Array of course ids



**General structure**

```

Default to "Array
(
)
" //Array of course ids
list of (
int //Course id
)

```

**XML-RPC (PHP structure)**

```

[courseids] =>
 Array
 (
 [0] => int
)

```

```
REST (POST parameters)
```

```
courseids[0]= int
```

Response

^

**General structure**

```

object {
workshops list of (
object {
id int //The primary key of the record.
course int //Course id this workshop is part of.
name string //Workshop name.
intro string //Workshop introduction text.
introformat int Default to "1" //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
lang string //Forced activity Language
instructauthors string Optional //Instructions for the submission phase.
instructauthorsformat int Default to "1" //instructauthors format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
instructreviewers string Optional //Instructions for the assessment phase.
instructreviewersformat int Default to "1" //instructreviewers format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
timemodified int Optional //The timestamp when the module was modified.
phase int Optional //The current phase of workshop (0 = not available, 1 = submission, 2 = assessment, 3 = closed).
useexamples int Optional //Optional feature: students practise evaluating on example submissions from teacher.
usepeerassessment int Optional //Optional feature: students perform peer assessment of others' work.
useselfassessment int Optional //Optional feature: students perform self assessment of their own work.
grade double Optional //The maximum grade for submission.
gradinggrade double Optional //The maximum grade for assessment.
strategy string Optional //The type of the current grading strategy used in this workshop.
evaluation string Optional //The recently used grading evaluation method.
gradedecimals int Optional //Number of digits that should be shown after the decimal point when displaying grades.
submissiontypetext int Optional //Indicates whether text is required as part of each submission. 0 for no, 1 for optional, 2 for required.
submissiontypefile int Optional //Indicates whether a file upload is required as part of each submission. 0 for no, 1 for optional, 2 for required.
nattachments int Optional //Maximum number of submission attachments.
submissionfiletypes string Optional //Comma separated List of file extensions.
latesubmissions int Optional //Allow submitting the work after the deadline.
maxbytes int Optional //Maximum size of the one attached file.
examplesmode int Optional //0 = example assessments are voluntary, 1 = examples must be assessed before submission,
 2 = examples are available after own submission and must be assessed before peer/self assessment phase.
submissionstart int Optional //0 = will be started manually, greater than 0 the timestamp of the start of the submission phase.
submissionend int Optional //0 = will be closed manually, greater than 0 the timestamp of the end of the submission phase.
assessmentstart int Optional //0 = will be started manually, greater than 0 the timestamp of the start of the assessment phase.
assessmentend int Optional //0 = will be closed manually, greater than 0 the timestamp of the end of the assessment phase.
phaseswitchassessment int Optional //Automatically switch to the assessment phase after the submissions deadline.
conclusion string Optional //A text to be displayed at the end of the workshop.
conclusionformat int Default to "1" //conclusion format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
overallfeedbackmode int Optional //Mode of the overall feedback support.
overallfeedbackfiles int Optional //Number of allowed attachments to the overall feedback.
overallfeedbackfiletypes string Optional //Comma separated list of file extensions.
overallfeedbackmaxbytes int Optional //Maximum size of one file attached to the overall feedback.
coursemodule int //coursemodule
introfiles //introfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)instructauthorsfiles Optional //instructauthorsfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)instructreviewersfiles Optional //instructreviewersfiles
list of (
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for the external files.
}
)conclusionfiles Optional //conclusionfiles
list of (

```

```
object {
 filename string Optional //File name.
 filepath string Optional //File path.
 filesize int Optional //File size.
 fileurl string Optional //Downloadable file url.
 timemodified int Optional //Time modified.
 mimetype string Optional //File mime type.
 isexternalfile int Optional //Whether is an external file.
 repositorytype string Optional //The repository type for the external files.
}
)
)}
)warnings Optional //List of warnings
list of (
 warning
object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
}
)
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [workshops] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [course] => int
 [name] => string
 [intro] => string
 [introformat] => int
 [lang] => string
 [instructauthors] => string
 [instructauthorsformat] => int
 [instructreviewers] => string
 [instructreviewersformat] => int
 [timemodified] => int
 [phase] => int
 [useexamples] => int
 [usepeerassessment] => int
 [useselfassessment] => int
 [grade] => double
 [gradinggrade] => double
 [strategy] => string
 [evaluation] => string
 [gradedecimals] => int
 [submissiontypetext] => int
 [submissiontypefile] => int
 [attachments] => int
 [submissionfiletypes] => string
 [latesubmissions] => int
 [maxbytes] => int
 [examplesmode] => int
 [submissionstart] => int
 [submissionend] => int
 [assessmentstart] => int
 [assessmentend] => int
 [phaseswitchassessment] => int
 [conclusion] => string
 [conclusionformat] => int
 [overallfeedbackmode] => int
 [overallfeedbackfiles] => int
 [overallfeedbackfiletypes] => string
 [overallfeedbackmaxbytes] => int
 [coursemodule] => int
 [introfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [instructauthorsfiles] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
 [instructreviewersfiles] =>
)
)
)
)
)
)
)
)

```



```
Array
(
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
[conclusionfiles] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="workshops">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="course">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="intro">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="introformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="instructauthors">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="instructauthorsformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="instructreviewers">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="instructreviewersformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="phase">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="useexamples">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usepeerassessment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="useselfassessment">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="gradinggrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="strategy">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="evaluation">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="gradedecimals">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissiontypefile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="nattachments">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="submissionfiletypes">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="latesubmissions">
 <VALUE>int</VALUE>
 </KEY>
```



```
<KEY name="maxbytes">
 <VALUE>int</VALUE>
</KEY>
<KEY name="examplesmode">
 <VALUE>int</VALUE>
</KEY>
<KEY name="submissionstart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="submissionend">
 <VALUE>int</VALUE>
</KEY>
<KEY name="assessmentstart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="assessmentend">
 <VALUE>int</VALUE>
</KEY>
<KEY name="phaseswitchassessment">
 <VALUE>int</VALUE>
</KEY>
<KEY name="conclusion">
 <VALUE>string</VALUE>
</KEY>
<KEY name="conclusionformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="overallfeedbackmode">
 <VALUE>int</VALUE>
</KEY>
<KEY name="overallfeedbackfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="overallfeedbackfiletypes">
 <VALUE>string</VALUE>
</KEY>
<KEY name="overallfeedbackmaxbytes">
 <VALUE>int</VALUE>
</KEY>
<KEY name="coursemodule">
 <VALUE>int</VALUE>
</KEY>
<KEY name="introfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="instructauthorsfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="fileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="mimetype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isexternalfile">
 <VALUE>int</VALUE>
</KEY>
<KEY name="repositorytype">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="instructreviewersfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
<KEY name="conclusionfiles">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</KEY>
<KEY name="warnings">
```



```

<MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### mod\_workshop\_update\_assessment

Add information to an allocated assessment.

### Arguments



**assessmentid** (Required)

Assessment id.

#### General structure

```
int //Assessment id.
```

#### XML-RPC (PHP structure)

```
[assessmentid] => int
```

#### REST (POST parameters)

```
assessmentid= int
```

**data** (Required)

Assessment data

**General structure**

```
//Assessment data
list of (
object {
name string //The assessment data (use WS get_assessment_form_definition for obtaining the data to sent).
 Apart from that data, you can optionally send:
 feedbackauthor (str); the feedback for the submission author
 feedbackauthorformat (int); the format of the feedbackauthor
 feedbackauthorinlineattachmentsid (int); the draft file area for the editor attachments
 feedbackauthorattachmentsid (int); the draft file area id for the feedback attachments
value string //The value of the option.
}
)
```

**XML-RPC (PHP structure)**

```
[data] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

**REST (POST parameters)**

```
data[0][name]= string
data[0][value]= string
```

**Response****General structure**

```
object {
status int //status: true if the assessment was added or updated false otherwise.
rawgrade double Optional //Raw percentual grade (0.00000 to 100.00000) for submission.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})
```

**XML-RPC (PHP structure)**

```
Array
(
[status] => int
[rawgrade] => double
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rawgrade">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_update\_submission**

Update the given submission.

**Arguments**

submissionid (Required)

Submission id

**General structure**

```
int //Submission id
```

**XML-RPC (PHP structure)**

```
[submissionid] => int
```

**REST (POST parameters)**

```
submissionid= int
```

**title** (Required)

Submission title

**General structure**

```
string //Submission title
```

**XML-RPC (PHP structure)**

```
[title] => string
```

**REST (POST parameters)**

```
title= string
```

**content** (Default to "")

Submission text content

**General structure**

```
string Default to "" //Submission text content
```

**XML-RPC (PHP structure)**

```
[content] => string
```

**REST (POST parameters)**

```
content= string
```

**contentformat** (Default to "0")

The format used for the content

**General structure**

```
int Default to "0" //The format used for the content
```

**XML-RPC (PHP structure)**

```
[contentformat] => int
```

**REST (POST parameters)**

```
contentformat= int
```

**inlineattachmentsid** (Default to "0")

The draft file area id for inline attachments in the content

**General structure**

```
int Default to "0" //The draft file area id for inline attachments in the content
```

**XML-RPC (PHP structure)**

```
[inlineattachmentsid] => int
```

**REST (POST parameters)**

```
inlineattachmentsid= int
```

**attachmentsid** (Default to "0")

The draft file area id for attachments

**General structure**

```
int Default to "0" //The draft file area id for attachments
```

**XML-RPC (PHP structure)**

```
[attachmentsid] => int
```

**REST (POST parameters)**

```
attachmentsid= int
```

**Response****General structure**

```
object {
 status int //True if the submission was updated false otherwise.
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_view\_submission**

Trigger the submission viewed event.

**Arguments****submissionid** (Required)

Submission id

**General structure**

int //Submission id

**XML-RPC (PHP structure)**

[submissionid] =&gt; int

**REST (POST parameters)**

```
submissionid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
})}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**mod\_workshop\_view\_workshop**

Trigger the course module viewed event and update the module completion status.

**Arguments****workshopid** (Required)

Workshop instance id

**General structure**

```
int //Workshop instance id
```

**XML-RPC (PHP structure)**

```
[workshopid] => int
```

**REST (POST parameters)**

```
workshopid= int
```

**Response****General structure**

```
object {
 status int //status: true if success
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [status] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

No

paygw\_paypal\_create\_transaction\_complete

Takes care of what needs to be done when a PayPal transaction comes back as complete.

## Arguments

component (Required)

The component name

### General structure

```
string //The component name
```

### XML-RPC (PHP structure)

```
[component] => string
```

### REST (POST parameters)

```
component= string
```

paymentarea (Required)

Payment area in the component

### General structure

```
string //Payment area in the component
```

### XML-RPC (PHP structure)

```
[paymentarea] => string
```

### REST (POST parameters)

```
paymentarea= string
```

itemid (Required)

The item id in the context of the component area

### General structure

```
int //The item id in the context of the component area
```

### XML-RPC (PHP structure)

```
[itemid] => int
```

### REST (POST parameters)

```
itemid= int
```

orderid (Required)

The order id coming back from PayPal

### General structure

```
string //The order id coming back from PayPal
```

### XML-RPC (PHP structure)

```
[orderid] => string
```

**REST (POST parameters)**

```
orderid= string
```

**Response****General structure**

```
object {
 success int //Whether everything was successful or not.
 message string //Message (usually the error message).
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [success] => int
 [message] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

**Callable from AJAX**

Yes

**paygw\_paypal\_get\_config\_for\_js**

Returns the configuration settings to be used in js

**Arguments**

**component** (Required)

Component

**General structure**

```
string //Component
```

**XML-RPC (PHP structure)**

```
[component] => string
```

**REST (POST parameters)**

```
component= string
```

**paymentarea (Required)**

Payment area in the component

**General structure**

```
string //Payment area in the component
```

**XML-RPC (PHP structure)**

```
[paymentarea] => string
```

**REST (POST parameters)**

```
paymentarea= string
```

**itemid (Required)**

An identifier for payment area in the component

**General structure**

```
int //An identifier for payment area in the component
```

**XML-RPC (PHP structure)**

```
[itemid] => int
```

**REST (POST parameters)**

```
itemid= int
```

**Response****General structure**

```
object {
 clientid string //PayPal client ID
 brandname string //Brand name
 cost double //Cost with gateway surcharge
 currency string //Currency
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [clientid] => string
 [brandname] => string
 [cost] => double
 [currency] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="clientid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="brandname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cost">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="currency">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**qbank\_columnsortorder\_set\_columnbank\_order**

Sets question columns order in database

**Arguments**

columns (Required)

**General structure**

```
list of (
 string //Plugin name for the column
)
```

**XML-RPC (PHP structure)**

```
[columns] =>
 Array
 (
 [0] => string
)
```

**REST (POST parameters)**

```
columns[0]= string
```

**Response****Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**qbank\_editquestion\_set\_status**

Update the question status.

**Arguments**

**questionid** (Required)

The question id

**General structure**

```
int //The question id
```

**XML-RPC (PHP structure)**

```
[questionid] => int
```

**REST (POST parameters)**

```
questionid= int
```

**status** (Required)

The updated question status

**General structure**

```
string //The updated question status
```

**XML-RPC (PHP structure)**

```
[status] => string
```

**REST (POST parameters)**

```
status= string
```

**Response****General structure**

```
object {
 status int //status: true if success
 statusname string //statusname: name of the status
 error string //Error message if error exists
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
 [statusname] => string
 [error] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="statusname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="error">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**qbank\_tagquestion\_submit\_tags\_form**

Update the question tags.

**Arguments****questionid** (Required)

The question id

**General structure**

```
int //The question id
```

**XML-RPC (PHP structure)**

```
[questionid] => int
```

**REST (POST parameters)**

```
questionid= int
```

**contextid** (Required)

The editing context id

**General structure**

```
int //The editing context id
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**formdata** (Required)

The data from the tag form

**General structure**

```
string //The data from the tag form
```

**XML-RPC (PHP structure)**

```
[formdata] => string
```

**REST (POST parameters)**

```
formdata= string
```

**Response**

**General structure**

```
object {
 status int //status: true if success
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [status] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**quizaccess\_seb\_validate\_quiz\_keys**

Validate a Safe Exam Browser config key or a browser exam key.

**Arguments****cmid** (Required)

Course module ID

**General structure**

```
int //Course module ID
```

**XML-RPC (PHP structure)**

```
[cmid] => int
```

**REST (POST parameters)**

cmid= int

**url** (Required)

Page URL to check

**General structure**

string //Page URL to check

**XML-RPC (PHP structure)**

[url] =&gt; string

**REST (POST parameters)**

url= string

**configkey** (Default to "null")

SEB config key

**General structure**

string Default to "null" //SEB config key

**XML-RPC (PHP structure)**

[configkey] =&gt; string

**REST (POST parameters)**

configkey= string

**browserexamkey** (Default to "null")

SEB browser exam key

**General structure**

string Default to "null" //SEB browser exam key

**XML-RPC (PHP structure)**

[browserexamkey] =&gt; string

**REST (POST parameters)**

browserexamkey= string

**Response****General structure**

```
object {
 configkey int //Is a provided config key valid?
 browserexamkey int //Is a provided browser exam key valid?
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [configkey] => int
 [browserexamkey] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="configkey">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="browserexamkey">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**report\_competency\_data\_for\_report**

Load the data for the competency report in a course.

**Arguments****courseid** (Required)

The course id

**General structure**

```
int //The course id
```

**XML-RPC (PHP structure)**

```
[courseid] => int
```

**REST (POST parameters)**

```
courseid= int
```

**userid** (Required)

The user id

**General structure**

```
int //The user id
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

moduleId (Required)

The module id

**General structure**

```
int //The module id
```

**XML-RPC (PHP structure)**

```
[moduleid] => int
```

**REST (POST parameters)**

```
moduleId= int
```

Response



```
General structure

object {
courseid int //Course id
user object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
usercompetencies list of (
object {
usercompetencycourse object {
userid int //userid
courseid int //courseid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
competency object {
linkedcourses //Linkedcourses
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
}
```

```
}

)relatedcompetencies //relatedcompetencies
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
)competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
framework object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //visible
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
contextid int //contextid
taxonomies string //taxonomies
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxonomyterm string //taxonomyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
```



```
pagecontextid int //pagecontextid
showlinks int //showlinks
}
pluginbaseurl string //pluginbaseurl
}
}
)pushratingstouserplans int //True if rating is push to user plans
}
```

^

**XML-RPC (PHP structure)**

```

Array
(
 [courseid] => int
 [user] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [course] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
 [usercompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [usercompetencycourse] =>
 Array
 (
 [userid] => int
 [courseid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
 [competency] =>
 Array
 (
 [linkedcourses] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
)
)
)
)
)
)

```



```
[summaryformat] => int
[startdate] => int
[enddate] => int
[visible] => int
[showactivitydates] => int
[showcompletionconditions] => int
[pdfexportfont] => string
[fullnamedisplay] => string
[viewurl] => string
[courseimage] => string
[progress] => int
[hasprogress] => int
[isfavourite] => int
[hidden] => int
[timeaccess] => int
[showshortname] => int
[coursecategory] => string
)
)
[relatedcompetencies] =>
Array
(
[0] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[sortorder] => int
[parentid] => int
[path] => string
[ruleoutcome] => int
[ruletype] => string
[ruleconfig] => string
[scaleid] => int
[scaleconfiguration] => string
[competencyframeworkid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
)
)
[competency] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[sortorder] => int
[parentid] => int
[path] => string
[ruleoutcome] => int
[ruletype] => string
[ruleconfig] => string
[scaleid] => int
[scaleconfiguration] => string
[competencyframeworkid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
)
)
[framework] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[visible] => int
[scaleid] => int
[scaleconfiguration] => string
[contextid] => int
[taxonomies] => string
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[canmanage] => int
[competenciescount] => int
)
```

```
[contextname] => string
[contextnamenoprefix] => string
)
[hascourses] => int
[hasrelatedcompetencies] => int
[scaleid] => int
[scaleconfiguration] => string
[taxonomyterm] => string
[comppath] =>
 Array
 (
 [ancestors] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
)
)
[framework] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
[pluginbaseurl] => string
[pagecontextid] => int
[showlinks] => int
)
[pluginbaseurl] => string
)
)
)
)
[pushratingstouserplans] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="user">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="course">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="pdfexportfont">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
 <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
 <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="usercompetencycourse">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="competency">
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
</KEY>
```



```
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="shortname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="summary">
 <VALUE>string</VALUE>
</KEY>
<KEY name="summaryformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="startdate">
 <VALUE>int</VALUE>
</KEY>
<KEY name="enddate">
 <VALUE>int</VALUE>
</KEY>
<KEY name="visible">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showactivitydates">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
 <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
 <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```
</KEY>
<KEY name="parentid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="path">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="hascourses">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="taxononyterm">
 <VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
```



```

<SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showlinks">
 <VALUE>int</VALUE>
 </KEY>
 <SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <SINGLE>
 </KEY>
 <KEY name="pushratingstouserplans">
 <VALUE>int</VALUE>
 </KEY>
</SINGLE>
</RESPONSE>

```



## Error message

### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

### report\_insights\_action\_executed

Stores an action executed over a group of predictions.

#### Arguments

actionname (Required)

The name of the action

##### General structure

```
string //The name of the action
```

##### XML-RPC (PHP structure)

```
[actionname] => string
```

##### REST (POST parameters)

```
actionname= string
```

predictionids (Required)

Array of prediction ids

##### General structure

```
//Array of prediction ids
list of (
int //Prediction id
)
```

##### XML-RPC (PHP structure)

```
[predictionids] =>
 Array
 (
 [0] => int
)

```



##### REST (POST parameters)

```
predictionids[0]= int
```

#### Response

**General structure**

```
object {
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**report\_insights\_set\_fixed\_prediction** (Deprecated)

Flags a prediction as fixed.

**Arguments****predictionid** (Required)

The prediction id

**General structure**

int //The prediction id

**XML-RPC (PHP structure)**

[predictionid] =&gt; int

**REST (POST parameters)**

predictionid= int

**Response****General structure**

```
object {
 success int //True if the prediction was successfully flagged as fixed.
 warnings Optional //List of warnings
 list of (
 /warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [success] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**report\_insights\_set\_notuseful\_prediction** (Deprecated)

Flags the prediction as not useful.

**Arguments****predictionid** (Required)

The prediction id

**General structure**

int //The prediction id

**XML-RPC (PHP structure)**

[predictionid] =&gt; int

**REST (POST parameters)**

```
predictionid= int
```

**Response****General structure**

```
object {
 success int //True if the prediction was successfully flagged as not useful.
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [success] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="success">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tiny\_autosave\_reset\_session**

Reset an autosave session

**Arguments****contextid** (Required)

The context id that owns the editor

**General structure**

```
int //The context id that owns the editor
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**pagehash** (Required)

The page hash

**General structure**

```
string //The page hash
```

**XML-RPC (PHP structure)**

```
[pagehash] => string
```

**REST (POST parameters)**

```
pagehash= string
```

**pageinstance** (Required)

The page instance

**General structure**

```
string //The page instance
```

**XML-RPC (PHP structure)**

```
[pageinstance] => string
```

**REST (POST parameters)**

```
pageinstance= string
```

**elementid (Required)**

The ID of the element

**General structure**

```
string //The ID of the element
```

**XML-RPC (PHP structure)**

```
[elementid] => string
```

**REST (POST parameters)**

```
elementid= string
```

**Response****General structure**

```
object {
}
```

**XML-RPC (PHP structure)**

```
Array
(
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tiny\_autosave\_resume\_session**

Resume an autosave session

**Arguments****contextid** (Required)

The context id that owns the editor

**General structure**

```
int //The context id that owns the editor
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**pagehash** (Required)

The page hash

**General structure**

```
string //The page hash
```

**XML-RPC (PHP structure)**

```
[pagehash] => string
```

**REST (POST parameters)**

```
pagehash= string
```

**pageinstance** (Required)

The page instance

**General structure**

```
string //The page instance
```

**XML-RPC (PHP structure)**

```
[pageinstance] => string
```

**REST (POST parameters)**

```
pageinstance= string
```

**elementid** (Required)

The ID of the element

**General structure**

```
string //The ID of the element
```

**XML-RPC (PHP structure)**

```
[elementid] => string
```

**REST (POST parameters)**

```
elementid= string
```

**draftid (Required)**

The new draft item id to resume files to

**General structure**

```
int //The new draft item id to resume files to
```

**XML-RPC (PHP structure)**

```
[draftid] => int
```

**REST (POST parameters)**

```
draftid= int
```

**Response****General structure**

```
object {
 drafttext string //The draft text
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [drafttext] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="drafttext">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tiny\_autosave\_update\_session**

Update an autosave session

**Arguments****contextid** (Required)

The context id that owns the editor

**General structure**

```
int //The context id that owns the editor
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**pagehash** (Required)

The page hash

**General structure**

```
string //The page hash
```

**XML-RPC (PHP structure)**

```
[pagehash] => string
```

**REST (POST parameters)**

```
pagehash= string
```

**pageinstance** (Required)

The page instance

**General structure**

```
string //The page instance
```

**XML-RPC (PHP structure)**

```
[pageinstance] => string
```

**REST (POST parameters)**

```
pageinstance= string
```

**elementid** (Required)

The ID of the element

**General structure**

```
string //The ID of the element
```

**XML-RPC (PHP structure)**

```
[elementid] => string
```

**REST (POST parameters)**

```
elementid= string
```

**drafttext** (Required)

The draft text

**General structure**

```
string //The draft text
```

**XML-RPC (PHP structure)**

```
[drafttext] => string
```

**REST (POST parameters)**

```
drafttext= string
```

**Response****General structure**

```
object {
}
```

**XML-RPC (PHP structure)**

```
Array
(
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tiny\_equation\_filter**

Filter the equation

**Arguments****contextid** (Required)

The context ID

**General structure**

```
int //The context ID
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**content** (Required)

The equation content

**General structure**

```
string //The equation content
```

**XML-RPC (PHP structure)**

```
[content] => string
```

**REST (POST parameters)**

```
content= string
```

**Response**

**General structure**

```
object {
content string //Filtered content
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [content] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_analytics\_potential\_contexts**

Retrieve the list of potential contexts for a model.

**Arguments****query** (Default to "null")

The model id

**General structure**

```
string Default to "null" //The model id
```

**XML-RPC (PHP structure)**

```
[query] => string
```

**REST (POST parameters)**

```
query= string
```

**modelid** (Default to "null")

The model id

**General structure**

```
int Default to "null" //The model id
```

**XML-RPC (PHP structure)**

```
[modelid] => int
```

**REST (POST parameters)**

```
modelid= int
```

**Response****General structure**

```
list of (
object {
id int //ID of the context
name string //The context name
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] =>
 Array
 (
 [id] => int
 [name] => string
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_behat\_get\_entity\_generator**

Get the generator details for an entity

**Arguments****entitytype** (Required)

Entity type that can be created by a generator.

**General structure**

```
string //Entity type that can be created by a generator.
```

**XML-RPC (PHP structure)**

```
[entitytype] => string
```

**REST (POST parameters)**

```
entitytype= string
```

**Response****General structure**

```
object {
required Optional //Required fields
list of (
string //Required field
)}
```

**XML-RPC (PHP structure)**

```
Array
(
[required] =>
 Array
 (
 [0] => string
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="required">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_approve\_data\_request**

Approve a data request

**Arguments****requestid** (Required)

The request ID

**General structure**

int //The request ID

**XML-RPC (PHP structure)**

[requestid] =&gt; int

**REST (POST parameters)**

requestid= int

**Response**

**General structure**

```

object {
 result int //The processing result
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

tool\_dataprivacy\_bulk\_approve\_data\_requests

Bulk approve data requests

#### Arguments

requestids (Required)

##### General structure

```
list of (
int //The request ID
)
```

##### XML-RPC (PHP structure)

```
[requestids] =>
Array
(
[0] => int
)
```

##### REST (POST parameters)

```
requestids[0]= int
```

#### Response

##### General structure

```
object {
result int //The processing result
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [result] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

Yes

## tool\_dataprivacy\_bulk\_deny\_data\_requests

## Bulk deny data requests

## Arguments

requestids (Required)

### General structure

```
list of (
int //The request ID
)
```

### XML-RPC (PHP structure)

```
[requestids] =>
Array
(
[0] => int
)
```

### REST (POST parameters)

```
requestids[0]= int
```

## Response

### General structure

```
object {
result int //The processing result
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)
}
```

### XML-RPC (PHP structure)

```
Array
(
[result] => int
[warnings] =>
Array
(
[0] =>
Array
(
[item] => string
[itemid] => int
[warningcode] => string
[message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_cancel\_data\_request**

Cancel the data request made by the user

**Arguments****requestid** (Required)

The request ID

**General structure**

```
int //The request ID
```

**XML-RPC (PHP structure)**

```
[requestid] => int
```

**REST (POST parameters)**

```
requestid= int
```

**Response****General structure**

```
object {
 result int //The processing result
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_confirm\_contexts\_for\_deletion**

Mark the selected expired contexts as confirmed for deletion

**Arguments****ids** (Default to "Array ()")

Array of expired context record IDs

**General structure**

```
Default to "Array
(
)
" //Array of expired context record IDs
list of (
int //Expired context record ID
)
```

**XML-RPC (PHP structure)**

```
[ids] =>
 Array
 (
 [0] => int
)
```

**REST (POST parameters)**

ids[0]= int

**Response****General structure**

```
object {
 result int //Whether the record was properly marked for deletion or not
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [result] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

Yes

## tool\_dataprivacy\_contact\_dpo

Contact the site Data Protection Officer(s)

## Arguments

### message (Required)

The user's message to the Data Protection Officer(s)

#### General structure

```
string //The user's message to the Data Protection Officer(s)
```

#### XML-RPC (PHP structure)

```
[message] => string
```

#### REST (POST parameters)

```
message= string
```

## Response

#### General structure

```
object {
 result int //The processing result
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

#### XML-RPC (PHP structure)

```
Array
(
 [
 [result] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_create\_category\_form****Adds a data category****Arguments****jsonformdata** (Required)

The data to create the category, encoded as a json array

**General structure**

```
string //The data to create the category, encoded as a json array
```

**XML-RPC (PHP structure)**

```
[jsonformdata] => string
```

## REST (POST parameters)

```
jsonformdata= string
```

## Response

```
General structure

object {
category object {
name string //The category name.
description string //The category description.
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
validationerrors int //Were there validation errors
warnings Optional //list of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}}
```

## XML-RPC (PHP structure)

```
Array
(
[category] =>
 Array
 (
 [name] => string
 [description] => string
 [descriptionformat] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
[validationerrors] => int
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="category">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="validationerrors">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_create\_purpose\_form**

Adds a data purpose

**Arguments****jsonformdata** (Required)

The data to create the purpose, encoded as a json array

**General structure**

```
string //The data to create the purpose, encoded as a json array
```

**XML-RPC (PHP structure)**

```
[jsonformdata] => string
```

**REST (POST parameters)**

```
jsonformdata= string
```

**Response**

```
General structure

object {
purpose object {
name string //The purpose name.
description string //The purpose description.
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
lawfulbases string //Comma-separated IDs matching records in tool_dataprivacy_lawfulbasis.
sensitizedatareasons string //Comma-separated IDs matching records in tool_dataprivacy_sensitive
retentionperiod string //Retention period. ISO_8601 durations format (as in DateInterval format).
protected int //Data retention with higher precedent over user's request to be forgotten.
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
formattedretentionperiod string //formattedretentionperiod
formattedlawfulbases //formattedlawfulbases
list of (
object {
name string //name
description string //description
})
)formattedsensitizedatareasons Optional //formattedsensitizedatareasons
list of (
object {
name string //name
description string //description
})
)roleoverrides string //roleoverrides
}
validationerrors int //Were there validation errors
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```

**XML-RPC (PHP structure)**

```
Array
(
 [purpose] =>
 Array
 (
 [name] => string
 [description] => string
 [descriptionformat] => int
 [lawfulbases] => string
 [sensitizeddatareasons] => string
 [retentionperiod] => string
 [protected] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [formattedretentionperiod] => string
 [formattedlawfulbases] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [description] => string
)
)
 [formattedsensitizeddatareasons] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [description] => string
)
)
)
 [roleoverrides] => string
)
[validationerrors] => int
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="purpose">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="lawfulbases">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="sensitizeddatareasons">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="retentionperiod">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="protected">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="formattedretentionperiod">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="formattedlawfulbases">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="formattedsensitizeddatareasons">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="roleoverrides">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="validationerrors">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```

</KEY>
<KEY name="warningcode">
 <VALUE>string</VALUE>
</KEY>
<KEY name="message">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

### tool\_dataprivacy\_delete\_category

Deletes an existing data category

#### Arguments

**id** (Required)

The category ID

#### General structure

```
int //The category ID
```



#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

### Response

**General structure**

```

object {
 result int //The processing result
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message****REST**

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`tool_dataprivacy_delete_purpose`

Deletes an existing data purpose

#### Arguments

`id` (Required)

The purpose ID

##### General structure

```
int //The purpose ID
```

##### XML-RPC (PHP structure)

```
[id] => int
```

##### REST (POST parameters)

```
id= int
```

#### Response

##### General structure

```
object {
 result int //The processing result
 warnings Optional //List of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```



##### XML-RPC (PHP structure)

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_deny\_data\_request**

Deny a data request

**Arguments****requestid** (Required)

The request ID

**General structure**

int //The request ID

**XML-RPC (PHP structure)**

[requestid] =&gt; int

**REST (POST parameters)**

```
requestid= int
```

**Response****General structure**

```
object {
 result int //The processing result
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

**XML-RPC (PHP structure)**

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_get\_activity\_options**

Fetches a list of activity options

**Arguments****nodefaults** (Default to "")

Whether to fetch all activities or only those without defaults

**General structure**

```
int Default to "" //Whether to fetch all activities or only those without defaults
```

**XML-RPC (PHP structure)**

```
[nodedefaults] => int
```

**REST (POST parameters)**

```
nodedefaults= int
```

**Response****General structure**

```
object {
 options list of (
 object {
 name string //The plugin name of the activity
 displayname string //The display name of the activity
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [options] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [displayname] => string
)
)
)
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="displayname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`tool_dataprivacy_get_category_options`

Fetches a list of data category options

**Arguments**

`includeinherit` (Default to "1")

Include option "Inherit"

**General structure**

```
int Default to "1" //Include option "Inherit"
```

**XML-RPC (PHP structure)**

```
[includeinherit] => int
```

**REST (POST parameters)**

```
includeinherit= int
```

`includenotset` (Default to "")

Include option "Not set"

**General structure**

```
int Default to "" //Include option "Not set"
```

**XML-RPC (PHP structure)**

```
[includenotset] => int
```

**REST (POST parameters)**

```
includenotset= int
```

**Response**

**General structure**

```
object {
options list of (
object {
id int //The category ID
name string //The category name
})
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
)}
```

**XML-RPC (PHP structure)**

```
Array
(
[options] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [name] => string
]
)
)
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
]
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_get\_data\_request**

Fetch the details of a user's data request

**Arguments****requestid** (Required)

The request ID

**General structure**

```
int //The request ID
```

**XML-RPC (PHP structure)**

```
[requestid] => int
```

**REST (POST parameters)**

```
requestid= int
```

**Response**

```

General structure

object {
 result object {
 type int //type
 comments string //comments
 commentsformat int //commentsformat
 userid int //userid
 requestedby int //requestedby
 status int //status
 dpo int //dpo
 dpocomment string //dpocomment
 dpocommentformat int //dpocommentformat
 systemapproved int //systemapproved
 creationmethod int //creationmethod
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 foruser object {
 id int //id
 email string //email
 idnumber string //idnumber
 phone1 string //phone1
 phone2 string //phone2
 department string //department
 institution string //institution
 fullname string //fullname
 identity string //identity
 profileurl string //profileurl
 profileimageurl string //profileimageurl
 profileimageurlsmall string //profileimageurlsmall
 }
 requestedbyuser object {
 id int //id
 email string //email
 idnumber string //idnumber
 phone1 string //phone1
 phone2 string //phone2
 department string //department
 institution string //institution
 fullname string //fullname
 identity string //identity
 profileurl string //profileurl
 profileimageurl string //profileimageurl
 profileimageurlsmall string //profileimageurlsmall
 }
 dpouser object {
 id int //id
 email string //email
 idnumber string //idnumber
 phone1 string //phone1
 phone2 string //phone2
 department string //department
 institution string //institution
 fullname string //fullname
 identity string //identity
 profileurl string //profileurl
 profileimageurl string //profileimageurl
 profileimageurlsmall string //profileimageurlsmall
 }
 messagehtml string Optional //messagehtml
 typename string //typename
 typenameshort string //typenameshort
 statuslabel string //statuslabel
 statuslabelclass string //statuslabelclass
 canreview int Optional //canreview
 approvedeny int Optional //approvedeny
 canmarkcomplete int Optional //canmarkcomplete
 }
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [result] =>
 Array
 (
 [type] => int
 [comments] => string
 [commentsformat] => int
 [userid] => int
 [requestedby] => int
 [status] => int
 [dpo] => int
 [dpocomment] => string
 [dpocommentformat] => int
 [systemapproved] => int
 [creationmethod] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [foruser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [requestedbyuser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [dpouser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [messagehtml] => string
 [typename] => string
 [typenameshort] => string
 [statuslabel] => string
 [statuslabelclass] => string
 [canreview] => int
 [approvedeny] => int
 [canmarkcomplete] => int
)
 [warnings] =>
 Array
 (

```

^

```
[0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <SINGLE>
 <KEY name="type">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="comments">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentsformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="requestedby">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="dpo">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="dpocomment">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="dpocommentformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="systemapproved">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="creationmethod">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="foruser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 </KEY>
 </RESPONSE>
```



```
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="requestedbyuser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="dpouser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
```



```

 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="messagehtml">
 <VALUE>string</VALUE>
</KEY>
<KEY name="typename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="typenameshort">
 <VALUE>string</VALUE>
</KEY>
<KEY name="statuslabel">
 <VALUE>string</VALUE>
</KEY>
<KEY name="statuslabelclass">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="approvedeny">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canmarkcomplete">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



#### Error message

##### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

#### Restricted to logged-in users

Yes

#### Callable from AJAX

Yes

#### tool\_dataprivacy\_get\_purpose\_options

Fetches a list of data storage purpose options

### Arguments

`includeinherit` (Default to "1")

Include option "Inherit"

#### General structure

```
int Default to "1" //Include option "Inherit"
```

#### XML-RPC (PHP structure)

```
[includeinherit] => int
```

#### REST (POST parameters)

```
includeinherit= int
```

`includenotset` (Default to "")

Include option "Not set"

#### General structure

```
int Default to "" //Include option "Not set"
```

#### XML-RPC (PHP structure)

```
[includenotset] => int
```

#### REST (POST parameters)

```
includenotset= int
```

### Response

#### General structure

```
object {
 options list of (
 object {
 id int //The purpose ID
 name string //The purpose name
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

## XML-RPC (PHP structure)

```
Array
(
 [options] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [name] => string
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="options">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_get\_users**

Fetches a list of users

**Arguments****query** (Required)

The search query

**General structure**

```
string //The search query
```

**XML-RPC (PHP structure)**

```
[query] => string
```

**REST (POST parameters)**

```
query= string
```

**Response****General structure**

```
list of (
object {
id int //ID of the user
fullname string //The fullname of the user
extrafields Optional //List of extra fields
list of (
object {
name string //Name of the extrafield.
value string //Value of the extrafield.
})
})
```

## XML-RPC (PHP structure)

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="extrafields">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</RESPONSE>
```



## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

Yes

tool\_dataprivacy\_mark\_complete

Mark a user's general enquiry as complete

### Arguments

**requestid** (Required)

The request ID

#### General structure

```
int //The request ID
```

#### XML-RPC (PHP structure)

```
[requestid] => int
```

#### REST (POST parameters)

```
requestid= int
```

### Response

#### General structure

```
object {
 result int //The processing result
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

#### XML-RPC (PHP structure)

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_set\_context\_defaults**

Updates the default category and purpose for a given context level (and optionally, a plugin)

**Arguments****contextlevel** (Required)

The context level

**General structure**

```
int //The context Level
```

**XML-RPC (PHP structure)**

```
[contextlevel] => int
```

**REST (POST parameters)**

```
contextlevel= int
```

**category (Required)**

The default category for the given context level

**General structure**

```
int //The default category for the given context level
```

**XML-RPC (PHP structure)**

```
[category] => int
```

**REST (POST parameters)**

```
category= int
```

**purpose (Required)**

The default purpose for the given context level

**General structure**

```
int //The default purpose for the given context level
```

**XML-RPC (PHP structure)**

```
[purpose] => int
```

**REST (POST parameters)**

```
purpose= int
```

**activity (Default to "null")**

The plugin name of the activity

**General structure**

```
string Default to "null" //The plugin name of the activity
```

**XML-RPC (PHP structure)**

```
[activity] => string
```

**REST (POST parameters)**

```
activity= string
```

**override (Default to "")**

Whether to override existing instances with the defaults

**General structure**

```
int Default to "" //Whether to override existing instances with the defaults
```

**XML-RPC (PHP structure)**

```
[override] => int
```

**REST (POST parameters)**

```
override= int
```

**Response****General structure**

```
object {
 result int //Whether the context defaults were successfully set or not
 warnings Optional //list of warnings
 list of (
 //warning
) object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
})}
```

**XML-RPC (PHP structure)**

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_set\_context\_form**

Sets purpose and category for a specific context

**Arguments****jsonformdata** (Required)

The context level data, encoded as a json array

**General structure**

```
string //The context level data, encoded as a json array
```

**XML-RPC (PHP structure)**

```
[jsonformdata] => string
```

**REST (POST parameters)**

```
jsonformdata= string
```

**Response****General structure**

```
object {
 result int //Whether the data was properly set or not
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [result] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

Yes

`tool_dataprivacy_set_contextlevel_form`

Sets purpose and category across a context level

## Arguments

`jsonformdata` (Required)

The context level data, encoded as a json array

### General structure

```
string //The context Level data, encoded as a json array
```

### XML-RPC (PHP structure)

```
[jsonformdata] => string
```

### REST (POST parameters)

```
jsonformdata= string
```

## Response

### General structure

```
object {
 result int //Whether the data was properly set or not
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

### XML-RPC (PHP structure)

```
Array
(
 [result] => int
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_dataprivacy\_tree\_extra\_branches**

Return branches for the context tree

**Arguments****contextid** (Required)

The context id to expand

**General structure**

```
int //The context id to expand
```

**XML-RPC (PHP structure)**

```
[contextid] => int
```

**REST (POST parameters)**

```
contextid= int
```

**element (Required)**

The element we are interested on

**General structure**

```
string //The element we are interested on
```

**XML-RPC (PHP structure)**

```
[element] => string
```

**REST (POST parameters)**

```
element= string
```

**Response****General structure**

```
object {
 branches list of (
 Optional //Node structure
 object {
 text string //The node text
 expandcontextid int //The contextid this node expands
 expandelement string //What element is this node expanded to
 contextid int //The node contextid
 contextlevel int //The node contextLevel
 expanded int //Is it expanded
 branches Optional //Children node structure
 list of (
 Optional //Node structure
 object {
 text string //The node text
 expandcontextid int //The contextid this node expands
 expandelement string //What element is this node expanded to
 contextid int //The node contextid
 contextlevel int //The node contextLevel
 expanded int //Is it expanded
 branches list of (
 string Optional //Nothing really, it will always be an empty array
)
 }
)
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [
 [branches] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [text] => string
 [expandcontextid] => int
 [expandelement] => string
 [contextid] => int
 [contextlevel] => int
 [expanded] => int
 [branches] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [text] => string
 [expandcontextid] => int
 [expandelement] => string
 [contextid] => int
 [contextlevel] => int
 [expanded] => int
 [branches] =>
 Array
 (
 [
 [0] => string
)
)
)
)
)
)
)
)
)
)
)
)
)
)
 [
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)
)
)
```



## REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="branches">
 <MULTIPLE>
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="expandcontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="expandelement">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextlevel">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="expanded">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="branches">
 <MULTIPLE>
 <SINGLE>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="expandcontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="expandelement">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextlevel">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="expanded">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="branches">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
</SINGLE>
</RESPONSE>

```



## Error message

### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

Yes

## tool\_lp\_data\_for\_competencies\_manage\_page

Load the data for the competencies manage page template

## Arguments

### competencyframeworkid (Required)

The competency framework id

#### General structure

```
int //The competency framework id
```

#### XML-RPC (PHP structure)

```
[competencyframeworkid] => int
```

#### REST (POST parameters)

```
competencyframeworkid= int
```

### search (Default to "")

A search string



#### General structure

```
string Default to "" //A search string
```

#### XML-RPC (PHP structure)

```
[search] => string
```

#### REST (POST parameters)

```
search= string
```

## Response

**General structure**

```

object {
framework object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //visible
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
contextid int //contextid
taxonomies string //taxonomies
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
canmanage int //True if this user has permission to manage competency frameworks
pagecontextid int //Context id for the framework
search string //Current search string
rulesmodules string //JSON encoded data for rules
pluginbaseurl string //Plugin base url
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [framework] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [visible] => int
 [scaleid] => int
 [scaleconfiguration] => string
 [contextid] => int
 [taxonomies] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [competenciescount] => int
 [contextname] => string
 [contextnamenoprefix] => string
]
)
 [canmanage] => int
 [pagecontextid] => int
 [search] => string
 [rulesmodules] => string
 [pluginbaseurl] => string
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="search">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="rulesmodules">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_lp\_data\_for\_competency\_frameworks\_manage\_page**

Load the data for the competency frameworks manage page template

**Arguments**

pagecontext (Required)

**General structure**

```
object {
 contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
 contextlevel string Default to "" //Context Level. To be used with instanceid.
 instanceid int Default to "0" //Context instance ID. To be used with level
}
```

**XML-RPC (PHP structure)**

```
[pagecontext] =>
 Array
 (
 [contextid] => int
 [contextlevel] => string
 [instanceid] => int
)
```

**REST (POST parameters)**

```
pagecontext[contextid]= int
pagecontext[contextlevel]= string
pagecontext[instanceid]= int
```

**Response**

**General structure**

```

object {
 competencyframeworks list of (
 object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 visible int //visible
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 contextid int //contextid
 taxonomies string //taxonomies
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 canmanage int //canmanage
 competenciescount int //competenciescount
 contextname string //contextname
 contextnamenoprefix string //contextnamenoprefix
 }
)pluginbaseurl string //Url to the tool_lp plugin folder on this Moodle site
 navigation list of (
 string //HTML for a navigation item that should be on this page
)pagecontextid int //The page context id
}

```

**XML-RPC (PHP structure)**

```

Array
(
 [competencyframeworks] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [visible] => int
 [scaleid] => int
 [scaleconfiguration] => string
 [contextid] => int
 [taxonomies] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [competenciescount] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
)
)
 [pluginbaseurl] => string
 [navigation] =>
 Array
 (
 [
 [0] => string
)
)
 [pagecontextid] => int
)

```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="competencyframeworks">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="navigation">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_lp\_data\_for\_competency\_summary**

Load competency data for summary template.

**Arguments****competencyid** (Required)

The competency id

**General structure**

```
int //The competency id
```

**XML-RPC (PHP structure)**

```
[competencyid] => int
```

**REST (POST parameters)**

```
competencyid= int
```

**includerelated** (Default to "")

Include or not related competencies

**General structure**

```
int Default to "" //Include or not related competencies
```

**XML-RPC (PHP structure)**

```
[includerelated] => int
```

**REST (POST parameters)**

```
includerelated= int
```

**includecourses** (Default to "")

Include or not competency courses

**General structure**

```
int Default to "" //Include or not competency courses
```

**XML-RPC (PHP structure)**

```
[includecourses] => int
```

**REST (POST parameters)**

```
includecourses= int
```

Response



```

General structure

object {
 linkedcourses //Linkedcourses
 list of (
 object {
 id int //id
 fullname string //fullname
 shortname string //shortname
 idnumber string //idnumber
 summary string //summary
 summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 startdate int //startdate
 enddate int //enddate
 visible int //visible
 showactivitydates int //showactivitydates
 showcompletionconditions int //showcompletionconditions
 pdfexportfont string //pdfexportfont
 fullnamedisplay string //fullnamedisplay
 viewurl string //viewurl
 courseimage string //courseimage
 progress int Optional //progress
 hasprogress int //hasprogress
 isfavourite int //isfavourite
 hidden int //hidden
 timeaccess int Optional //timeaccess
 showshortname int //showshortname
 coursecategory string //coursecategory
 }
)relatedcompetencies //relatedcompetencies
 list of (
 object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
)competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
 framework object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 visible int //visible
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 contextid int //contextid
 taxonomies string //taxonomies
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
}

```

```
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxonomyterm string //taxononyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showLinks
}
pluginbaseurl string //pluginbaseurl
}
```



**XML-RPC (PHP structure)**

```

Array
(
 [linkedcourses] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
 [relatedcompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)
 [competency] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)

```

^

```
(
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [visible] => int
 [scaleid] => int
 [scaleconfiguration] => string
 [contextid] => int
 [taxonomies] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [competenciescount] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
)
[hascourses] => int
[hasrelatedcompetencies] => int
[scaleid] => int
[scaleconfiguration] => string
[taxononyterm] => string
[comppath] =>
 Array
 (
 [
 [ancestors] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
 [framework] =>
 Array
 (
 [
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
 [pluginbaseurl] => string
 [pagecontextid] => int
 [showlinks] => int
)
 [pluginbaseurl] => string
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pdlexportfont">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasprogress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfavourite">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeaccess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showshortname">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

```
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="description">
 <VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="sortorder">
 <VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="path">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="competency">
<SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
</SINGLE>
```



```
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="hascourses">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
 <VALUE>int</VALUE>
</KEY>
```



```
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="taxononyterm">
 <VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
 <SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showlinks">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>
```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`tool_lp_data_for_course_competencies_page`

Load the data for the course competencies page template.

**Arguments**

`courseid` (Required)

The course id

**General structure**

```
int //The course id
```

**XML-RPC (PHP structure)**

```
[courseid] => int
```

**REST (POST parameters)**

```
courseid= int
```

`moduleid` (Default to "0")

The module id

**General structure**

```
int Default to "0" //The module id
```

**XML-RPC (PHP structure)**

```
[moduleid] => int
```

**REST (POST parameters)**

```
moduleid= int
```

**Response**

## General structure

```
url string Optional //url
iconurl string //iconurl
}
)usercompetencycourse object {
userid int //userid
courseid int //courseid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
ruleoutcomeoptions list of (
object {
value int //The option value
text string //The name of the option
selected int //If this is the currently selected option
}
)comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showLinks
}
plans list of (
object {
name string //name
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
userid int //userid
templateid int //templateid
origtemplateid int //origtemplateid
status int //status
duedate int //duedate
reviewerid int //reviewerid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
statusname string //statusname
isbasedontemplate int //isbasedontemplate
canmanage int //canmanage
canrequestreview int //canrequestreview
canreview int //canreview
canbeedited int //canbeedited
isactive int //isactive
isdraft int //isdraft
iscompleted int //iscompleted
isinreview int //isinreview
iswaitingforreview int //iswaitingforreview
isreopnallowed int //isreopnallowed
iscompleteallowed int //iscompleteallowed
isunlinkallowed int //isunlinkallowed
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isapproveallowed int //isapproveallowed
isunapproveallowed int //isunapproveallowed
duedateformatted string //duedateformatted
commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
}
```



```
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
template object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
})
)manageurl string //Url to the manage competencies page.
pluginbaseurl string //Url to the course competencies page.
}
```

**XML-RPC (PHP structure)**

```

Array
(
 [courseid] => int
 [pagecontextid] => int
 [gradableuserid] => int
 [canmanagecompetencyframeworks] => int
 [canmanagecoursecompetencies] => int
 [canconfigurecoursecompetencies] => int
 [cangradecompetencies] => int
 [settings] =>
 Array
 (
 [courseid] => int
 [pushratingstouserplans] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
 [statistics] =>
 Array
 (
 [competencycount] => int
 [proficientcompetencycount] => int
 [proficientcompetencypercentage] => double
 [proficientcompetencypercentageformatted] => string
 [leastproficient] =>
 Array
 (
 [0] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
 [leastproficientcount] => int
 [canbegradedincourse] => int
 [canmanagecoursecompetencies] => int
)
 [competencies] =>
 Array
 (
 [0] =>
 Array
 (
 [competency] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
)
)
)
)

```



```
[usermodified] => int
)
[coursecompetency] =>
 Array
 (
 [courseid] => int
 [competencyid] => int
 [sortorder] => int
 [ruleoutcome] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
[coursemodules] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [url] => string
 [iconurl] => string
)
)
[usercompetencycourse] =>
 Array
 (
 [userid] => int
 [courseid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
[ruleoutcomeoptions] =>
 Array
 (
 [0] =>
 Array
 (
 [value] => int
 [text] => string
 [selected] => int
)
)
[comppath] =>
 Array
 (
 [ancestors] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
)
[framework] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
[pluginbaseurl] => string
[pagecontextid] => int
[showlinks] => int
)
[plans] =>
 Array
```



```
(
[0] =>
 Array
 (
 [name] => string
 [description] => string
 [descriptionformat] => int
 [userid] => int
 [templatetid] => int
 [origtemplateid] => int
 [status] => int
 [duedate] => int
 [reviewerid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [statusname] => string
 [isbasedontemplate] => int
 [canmanage] => int
 [canrequestreview] => int
 [canreview] => int
 [canbeedited] => int
 [isactive] => int
 [isdraft] => int
 [iscompleted] => int
 [isinreview] => int
 [iswaitingforreview] => int
 [isreopenallowed] => int
 [iscompleteallowed] => int
 [isunlinkallowed] => int
 [isrequestreviewallowed] => int
 [iscancelreviewrequestallowed] => int
 [issstartreviewallowed] => int
 [issstopreviewallowed] => int
 [isapproveallowed] => int
 [isunapproveallowed] => int
 [duedateformatted] => string
 [commentarea] =>
 Array
 (
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
 [reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [template] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
)
)
)
```



```
[visible] => int
[contextid] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
)
)
)
[manageurl] => string
[pluginbaseurl] => string
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradableuserid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanagecompetencyframeworks">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanagecoursecompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canconfigurecoursecompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cangradecompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="settings">
 <SINGLE>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pushratingstouserplans">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="statistics">
 <SINGLE>
 <KEY name="competencycount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficientcompetencycount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficientcompetencypercentage">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="proficientcompetencypercentageformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="leastproficient">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="path">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="leastproficientcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canbegradedincourse">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canmanagecoursecompetencies">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="competencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
</KEY>
<KEY name="coursecompetency">
 <SINGLE>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="coursemodules">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="iconurl">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="usercompetencycourse">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcomeoptions">
 <MULTIPLE>
 <SINGLE>
 <KEY name="value">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="text">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="selected">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="comppath">
 <SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="showlinks">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="plans">
<MULTIPLE>
<SINGLE>
<KEY name="name">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="templatelid">
<VALUE>int</VALUE>
</KEY>
<KEY name="origtemplateid">
<VALUE>int</VALUE>
</KEY>
<KEY name="status">
<VALUE>int</VALUE>
</KEY>
<KEY name="duedate">
<VALUE>int</VALUE>
</KEY>
<KEY name="reviewerid">
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="statusname">
<VALUE>string</VALUE>
</KEY>
<KEY name="isbasedontemplate">
<VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
<VALUE>int</VALUE>
</KEY>
<KEY name="canrequestreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="canbeedited">
<VALUE>int</VALUE>
</KEY>
<KEY name="isactive">
<VALUE>int</VALUE>
</KEY>
<KEY name="isdraft">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscompleted">
<VALUE>int</VALUE>
</KEY>
<KEY name="isinreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="iswaitingforreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="isreopenallowed">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="iscompleteallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isunlinkallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
<SINGLE>
<KEY name="component">
<VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
<VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
<VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="cid">
<VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
<VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
<VALUE>int</VALUE>
</KEY>
<KEY name="canview">
<VALUE>int</VALUE>
</KEY>
<KEY name="count">
<VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
<VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
<VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
<VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
<VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
<VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
<VALUE>int</VALUE>
</KEY>
<KEY name="template">
<VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
```



```
<VALUE>int</VALUE>
</KEY>
</SINGLE>
<KEY name="reviewer">
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="email">
<VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
<VALUE>string</VALUE>
</KEY>
<KEY name="phone2">
<VALUE>string</VALUE>
</KEY>
<KEY name="department">
<VALUE>string</VALUE>
</KEY>
<KEY name="institution">
<VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
<VALUE>string</VALUE>
</KEY>
<KEY name="identity">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
<VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="template">
<SINGLE>
<KEY name="shortname">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="duedate">
<VALUE>int</VALUE>
</KEY>
<KEY name="visible">
<VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="cohortscount">
<VALUE>int</VALUE>
</KEY>
<KEY name="planscount">
```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canread">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="manageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

#### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

#### Restricted to logged-in users

Yes



#### Callable from AJAX

Yes

[tool\\_lp\\_data\\_for\\_plan\\_page](#)

Load the data for the plan page template.

#### Arguments

[planid](#) (Required)

The plan id

#### General structure

```
int //The plan id
```

**XML-RPC (PHP structure)**

```
[planid] => int
```

**REST (POST parameters)**

```
planid= int
```

Response



```

General structure

object {
plan object {
name string //name
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
userid int //userid
templateid int //templateid
origtemplateid int //origtemplateid
status int //status
duedate int //duedate
reviewerid int //reviewerid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
statusname string //statusname
isbasedontemplate int //isbasedontemplate
canmanage int //canmanage
canrequestreview int //canrequestreview
canreview int //canreview
canbeedited int //canbeedited
isactive int //isactive
isdraft int //isdraft
iscompleted int //iscompleted
isinreview int //isinreview
iswaitingforreview int //iswaitingforreview
isreopnallowed int //isreopnallowed
iscompleteallowed int //iscompleteallowed
isunlinkallowed int //isunlinkallowed
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
issstopreviewallowed int //issstopreviewallowed
isapproveallowed int //isapproveallowed
isunapproveallowed int //isunapproveallowed
duedateformatted string //duedateformatted
commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
template object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
}

```



```
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
contextid int //Context ID.
pluginbaseurl string //Plugin base URL.
competencies list of (
object {
competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showLinks
}
usercompetency object {
userid int //userid
competencyid int //competencyid
status int //status
reviewerid int //reviewerid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canrequestreview int //canrequestreview
canreview int //canreview
gradename string //gradename
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
isstartreviewallowed int //isstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isstatusidle int //isstatusidle
isstatusinreview int //isstatusinreview
isstatuswaitingforreview int //isstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
}
```

```
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
usercompetencyplan object {
userid int //userid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
planid int //planid
sortorder int //sortorder
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
}
)competencycount int //Count of competencies
proficientcompetencycount int //Count of proficientcompetencies
proficientcompetencypercentage double //Percentage of competencies proficient
proficientcompetencypercentageformatted string //Displayable percentage
}
```

^

## XML-RPC (PHP structure)

```

 [
 [plan] =>
 Array
 (
 [
 [name] => string
 [description] => string
 [descriptionformat] => int
 [userid] => int
 [templateid] => int
 [origtemplateid] => int
 [status] => int
 [duedate] => int
 [reviewerid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [statusname] => string
 [isbasedontemplate] => int
 [canmanage] => int
 [canrequestreview] => int
 [canreview] => int
 [canbeedited] => int
 [isactive] => int
 [isdraft] => int
 [iscompleted] => int
 [isinreview] => int
 [iswaitingforreview] => int
 [isreopenable] => int
 [incompleteallowed] => int
 [isunlinkallowed] => int
 [isrequestreviewallowed] => int
 [iscancelreviewrequestallowed] => int
 [isstartreviewallowed] => int
 [istopreviewallowed] => int
 [isapproveallowed] => int
 [isunapproveallowed] => int
 [duedateformatted] => string
 [commentarea] =>
 Array
 (
 [
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
)
)
 [reviewer] =>
 Array
 (
 [
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
)
)
 [template] =>
 Array
 (

```

```

[shortname] => string
[description] => string
[descriptionformat] => int
[duedate] => int
[visible] => int
[contextid] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[duedateformatted] => string
[cohortscount] => int
[planscount] => int
[canmanage] => int
[canread] => int
[contextname] => string
[contextnamenoprefix] => string
)
[url] => string
)
[contextid] => int
[pluginbaseurl] => string
[competencies] =>
 Array
 (
 [0] =>
 Array
 (
 [competency] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timemcreated] => int
 [timemmodified] => int
 [usermodified] => int
)
 [comppath] =>
 Array
 (
 [ancestors] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
)
 [framework] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
 [pluginbaseurl] => string
 [pagecontextid] => int
 [showlinks] => int
)
 [usercompetency] =>
 Array
 (
 [userid] => int
 [competencyid] => int
)
)
)
)
)

```



```
[status] => int
[reviewerid] => int
[proficiency] => int
[grade] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[canrequestreview] => int
[canreview] => int
[gradename] => string
[isrequestreviewallowed] => int
[iscancelreviewrequestallowed] => int
[isstartreviewallowed] => int
[isstopreviewallowed] => int
[isstatusidle] => int
[isstatusinreview] => int
[isstatuswaitingforreview] => int
[proficiencyname] => string
[reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [statusname] => string
 [url] => string
)
)
[usercompetencyplan] =>
 Array
 (
 [userid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [planid] => int
 [sortorder] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
)
)
)
[competencycount] => int
[proficientcompetencycount] => int
[proficientcompetencypercentage] => double
[proficientcompetencypercentageformatted] => string
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="plan">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="templateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="origtemplateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="statusname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isbasedontemplate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canbeedited">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isactive">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isdraft">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscompleted">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isinreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iswaitingforreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isreopenallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscompleteallowed">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="isunlinkallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="autostart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canpost">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="collapsediconkey">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="displaytotalcount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displaycancel">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullwidth">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="linktext">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="notoggle">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="template">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>

```



```
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="template">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="canread">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="comppath">
```



```
<SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showlinks">
 <VALUE>int</VALUE>
 </KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="canrequestreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencyplan">
```



```

<SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="competencycount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficientcompetencycount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficientcompetencypercentage">
 <VALUE>double</VALUE>
</KEY>
<KEY name="proficientcompetencypercentageformatted">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

## tool\_lp\_data\_for\_plans\_page

Load the data for the plans page template

### Arguments

userid (Required)

The user id

#### General structure

```
int //The user id
```

#### XML-RPC (PHP structure)

```
[userid] => int
```

#### REST (POST parameters)

```
userid= int
```

### Response



**General structure**

```

object {
 userid int //The Learning plan user id
 plans list of (
 object {
 name string //name
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 userid int //userid
 templateid int //templateid
 origtemplateid int //origtemplateid
 status int //status
 duedate int //duedate
 reviewerid int //reviewerid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 statusname string //statusname
 isbasedontemplate int //isbasedontemplate
 canmanage int //canmanage
 canrequestreview int //canrequestreview
 canreview int //canreview
 canbeedited int //canbeedited
 isactive int //isactive
 isdraft int //isdraft
 iscompleted int //iscompleted
 isinreview int //isinreview
 iswaitingforreview int //iswaitingforreview
 isreopenallowed int //isreopenallowed
 iscompleteallowed int //iscompleteallowed
 isunlinkallowed int //isunlinkallowed
 isrequestreviewallowed int //isrequestreviewallowed
 iscancelreviewrequestallowed int //iscancelreviewrequestallowed
 isstartreviewallowed int //isstartreviewallowed
 isstopreviewallowed int //isstopreviewallowed
 isapprovewallowed int //isapprovewallowed
 isunapprovewallowed int //isunapprovewallowed
 duedateformatted string //duedateformatted
 commentarea object {
 component string //component
 commentarea string //commentarea
 itemid int //itemid
 courseid int //courseid
 contextid int //contextid
 cid string //cid
 autostart int //autostart
 canpost int //canpost
 canview int //canview
 count int //count
 collapsediconkey string //collapsediconkey
 displaytotalcount int //displaytotalcount
 displaycancel int //displaycancel
 fullwidth int //fullwidth
 linktext string //linktext
 notoggle int //notoggle
 template string //template
 canpostorhascomments int //canpostorhascomments
 }
 reviewer object {
 id int //id
 email string //email
 idnumber string //idnumber
 phone1 string //phone1
 phone2 string //phone2
 department string //department
 institution string //institution
 fullname string //fullname
 identity string //identity
 profileurl string //profileurl
 profileimageurl string //profileimageurl
 profileimageurlsmall string //profileimageurlsmall
 }
 template object {
 shortname string //shortname
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 duedate int //duedate
 visible int //visible
 contextid int //contextid
 id int //id
 timecreated int //timecreated
 }
 }
)
}

```

```
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
)pluginbaseurl string //Url to the tool_lp plugin folder on this Moodle site
navigation list of (
string //HTML for a navigation item that should be on this page
)canreadusevidence int //Can the current user view the user's evidence
canmanageuserplans int //Can the current user manage the user's plans
}
```



**XML-RPC (PHP structure)**

```

Array
(
 [userid] => int
 [plans] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [name] => string
 [description] => string
 [descriptionformat] => int
 [userid] => int
 [templatetid] => int
 [origtemplateid] => int
 [status] => int
 [duedate] => int
 [reviewerid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [statusname] => string
 [isbasedontemplate] => int
 [canmanage] => int
 [canrequestreview] => int
 [canreview] => int
 [canbeedited] => int
 [isactive] => int
 [isdraft] => int
 [iscompleted] => int
 [isinreview] => int
 [iswaitingforreview] => int
 [isreopenallowed] => int
 [iscompleteallowed] => int
 [isunlinkallowed] => int
 [isrequestreviewallowed] => int
 [iscancelreviewrequestallowed] => int
 [isstartreviewallowed] => int
 [issstopreviewallowed] => int
 [isapproveallowed] => int
 [isunapproveallowed] => int
 [duedateformatted] => string
 [commentarea] =>
 Array
 (
 [
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
)
 [reviewer] =>
 Array
 (
 [
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
)
)
)
)
)
)

```



```
)
 [template] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
 [visible] => int
 [contextid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [duedateformatted] => string
 [cohortscount] => int
 [planscount] => int
 [canmanage] => int
 [canread] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
 [url] => string
)
)
)
[pluginbaseurl] => string
[navigation] =>
 Array
 (
 [0] => string
)
[canreaduserevidence] => int
[canmanageuserplans] => int
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="plans">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="templateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="origtemplateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="statusname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isbasedontemplate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canbeedited">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isactive">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isdraft">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscompleted">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isinreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iswaitingforreview">
 <VALUE>int</VALUE>
 </KEY>
```



```
<KEY name="isreopenallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscompleteallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isunlinkallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="issstartreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cid">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="autostart">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canpost">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="collapsediconkey">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="displaytotalcount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="displaycancel">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullwidth">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="linktext">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="notoggle">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="template">
 <VALUE>string</VALUE>
 </KEY>
```



```
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="template">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```

<KEY name="planscount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canread">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="navigation">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
</KEY>
<KEY name="canreaduserevidence">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canmanageuserplans">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

### tool\_lp\_data\_for\_related\_competencies\_section

Load the data for the related competencies template.

### Arguments

**competencyid** (Required)

The competency id

```
General structure

int //The competency id
```

```
XML-RPC (PHP structure)

[competencyid] => int
```

```
REST (POST parameters)

competencyid= int
```

## Response

```
General structure

object {
relatedcompetencies list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
)showdeleterelatedaction int //Whether to show the delete relation link or not
}
```



**XML-RPC (PHP structure)**

```
Array
(
 [relatedcompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)
 [showdeleterelatedaction] => int
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="showdeleterelatedaction">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_lp\_data\_for\_template\_competencies\_page**

Load the data for the template competencies page template.

**Arguments****templateid** (Required)

The template id

**General structure**

```
int //The template id
```

**XML-RPC (PHP structure)**

```
[templateid] => int
```

**REST (POST parameters)**

```
templateid= int
```

**pagecontext** (Required)**General structure**

```
object {
 contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
 contextlevel string Default to "" //Context Level. To be used with instanceid.
 instanceid int Default to "0" //Context instance ID. To be used with level
}
```

**XML-RPC (PHP structure)**

```
[pagecontext] =>
 Array
 (
 [contextid] => int
 [contextlevel] => string
 [instanceid] => int
)
```

**REST (POST parameters)**

```
pagecontext[contextid]= int
pagecontext[contextlevel]= string
pagecontext[instanceid]= int
```

## Response

^

```
General structure

object {
 template object {
 shortname string //shortname
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 duedate int //duedate
 visible int //visible
 contextid int //contextid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 duedateformatted string //duedateformatted
 cohortscount int //cohortscount
 planscount int //planscount
 canmanage int //canmanage
 canread int //canread
 contextname string //contextname
 contextnamenoprefix string //contextnamenoprefix
 }
 pagecontextid int //Context ID
 canmanageteamcompetencies int //User can manage competency frameworks
 canmanagetemplatecompetencies int //User can manage Learning plan templates
 competencies list of (
 object {
 linkedcourses //Linkedcourses
 list of (
 object {
 id int //id
 fullname string //fullname
 shortname string //shortname
 idnumber string //idnumber
 summary string //summary
 summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 startdate int //startdate
 enddate int //enddate
 visible int //visible
 showactivitydates int //showactivitydates
 showcompletionconditions int //showcompletionconditions
 pdfexportfont string //pdfexportfont
 fullnamedisplay string //fullnamedisplay
 viewurl string //viewurl
 courseimage string //courseimage
 progress int Optional //progress
 hasprogress int //hasprogress
 isfavourite int //isfavourite
 hidden int //hidden
 timeaccess int Optional //timeaccess
 showshortname int //showshortname
 coursecategory string //coursecategory
 }
)relatedcompetencies //relatedcompetencies
 list of (
 object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
)competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 }
 }
)
}
```

```
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
framework object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //visible
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
contextid int //contextid
taxonomies string //taxonomies
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxononyterm string //taxononyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showLinks
}
pluginbaseurl string //pluginbaseurl
)
manageurl string //Url to the manage competencies page.
pluginbaseurl string //Base URL of the plugin.
statistics object {
competencycount int //competencycount
unlinkedcompetencycount int //unlinkedcompetencycount
plancount int //plancount
completedplancount int //completedplancount
usercompetencyplancount int //usercompetencyplancount
proficientusercompetencyplancount int //proficientusercompetencyplancount
linkedcompetencypercentage double //linkedcompetencypercentage
linkedcompetencypercentageformatted string //linkedcompetencypercentageformatted
linkedcompetencycount int //linkedcompetencycount
completedplanpercentage double //completedplanpercentage
completedplanpercentageformatted string //completedplanpercentageformatted
proficientusercompetencyplanpercentage double //proficientusercompetencyplanpercentage
proficientusercompetencyplanpercentageformatted string //proficientusercompetencyplanpercentageformatted
leastproficient //leastproficient
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
}
```

```
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
leastproficientcount int //Leastproficientcount
}
}
```

^

**XML-RPC (PHP structure)**

```

Array
(
 [template] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
 [visible] => int
 [contextid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [duedateformatted] => string
 [cohortscount] => int
 [planscount] => int
 [canmanage] => int
 [canread] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
 [pagecontextid] => int
 [canmanagecompetencyframeworks] => int
 [canmanageteplatecompetencies] => int
 [competencies] =>
 Array
 (
 [0] =>
 Array
 (
 [linkedcourses] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
)
 [relatedcompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
)
)
)
)
)
)
)
)
)

```



```

[competencyframeworkid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
)
)
)
[competency] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[sortorder] => int
[parentid] => int
[path] => string
[ruleoutcome] => int
[rulename] => string
[ruleconfig] => string
[scaleid] => int
[scaleconfiguration] => string
[competencyframeworkid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
)
)
[framework] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[visible] => int
[scaleid] => int
[scaleconfiguration] => string
[contextid] => int
[taxonomies] => string
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[cannemanage] => int
[competenciescount] => int
[contextname] => string
[contextnamenoprefix] => string
)
)
[hascourses] => int
[hasrelatedcompetencies] => int
[scaleid] => int
[scaleconfiguration] => string
[taxonomyterm] => string
[comppath] =>
Array
(
[ancestors] =>
Array
(
[0] =>
Array
(
[id] => int
[name] => string
[first] => int
[last] => int
[position] => int
)
)
)
[framework] =>
Array
(
[id] => int
[name] => string
[first] => int
[last] => int
[position] => int
)
)
[pluginbaseurl] => string
[pagecontextid] => int
[showlinks] => int
)
)

```



```
[pluginbaseurl] => string
)
)
[manageurl] => string
[pluginbaseurl] => string
[statistics] =>
Array
(
[competencycount] => int
[unlinkedcompetencycount] => int
[plancount] => int
[completedplancount] => int
[usercompetencyplancount] => int
[proficientusercompetencyplancount] => int
[linkedcompetencypercentage] => double
[linkedcompetencypercentageformatted] => string
[linkedcompetencycount] => int
[completedplanpercentage] => double
[completedplanpercentageformatted] => string
[proficientusercompetencyplanpercentage] => double
[proficientusercompetencyplanpercentageformatted] => string
[leastproficient] =>
Array
(
[0] =>
Array
(
[shortname] => string
[idnumber] => string
[description] => string
[descriptionformat] => int
[sortorder] => int
[parentid] => int
[path] => string
[ruleoutcome] => int
[ruletype] => string
[ruleconfig] => string
[scaleid] => int
[scaleconfiguration] => string
[competencyframeworkid] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
)
)
[leastproficientcount] => int
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="template">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canread">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanagecompetencyframeworks">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanagetemplatecompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="summary">
 <VALUE>string</VALUE>
</KEY>
<KEY name="summaryformat">
 <VALUE>int</VALUE>
</KEY>
<KEY name="startdate">
 <VALUE>int</VALUE>
</KEY>
<KEY name="enddate">
 <VALUE>int</VALUE>
</KEY>
<KEY name="visible">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showactivitydates">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
 <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
 <VALUE>int</VALUE>
</KEY>
<KEY name="coursecategory">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
<SINGLE>
<KEY name="shortname">
<VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="visible">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
<VALUE>string</VALUE>
</KEY>
<KEY name="contextid">
<VALUE>int</VALUE>
</KEY>
<KEY name="taxonomies">
<VALUE>string</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
<VALUE>int</VALUE>
</KEY>
<KEY name="competenciescount">
<VALUE>int</VALUE>
</KEY>
<KEY name="contextname">
<VALUE>string</VALUE>
</KEY>
<KEY name="contextnamenoprefix">
<VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="hascourses">
<VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
<VALUE>string</VALUE>
</KEY>
<KEY name="taxonomyterm">
<VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
<SINGLE>
<KEY name="ancestors">
<MULTIPLE>
<SINGLE>
<KEY name="id">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="name">
 <VALUE>string</VALUE>
</KEY>
<KEY name="first">
 <VALUE>int</VALUE>
</KEY>
<KEY name="last">
 <VALUE>int</VALUE>
</KEY>
<KEY name="position">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showlinks">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <SINGLE>
 <KEY>
 <KEY name="manageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 </KEY>
 <KEY name="statistics">
 <SINGLE>
 <KEY name="competencycount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="unlinkedcompetencycount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="plancount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="completedplancount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usercompetencyplancount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficientusercompetencyplancount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="linkedcompetencypercentage">
 <VALUE>double</VALUE>
 </KEY>
 <KEY name="linkedcompetencypercentageformatted">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="linkedcompetencycount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="completedplanpercentage">
 <VALUE>double</VALUE>
</KEY>
<KEY name="completedplanpercentageformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="proficientusercompetencyplanpercentage">
 <VALUE>double</VALUE>
</KEY>
<KEY name="proficientusercompetencyplanpercentageformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="leastproficient">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="leastproficientcount">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
</RESPONSE>
```



### Error message

#### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

`tool_lp_data_for_templates_manage_page`

Load the data for the learning plan templates manage page template

### Arguments

`pagecontext` (Required)

#### General structure

```
object {
 contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
 contextlevel string Default to "" //Context Level. To be used with instanceid.
 instanceid int Default to "0" //Context instance ID. To be used with level
}
```

#### XML-RPC (PHP structure)

```
[pagecontext] =>
 Array
 (
 [contextid] => int
 [contextlevel] => string
 [instanceid] => int
)
```



#### REST (POST parameters)

```
pagecontext[contextid]= int
pagecontext[contextlevel]= string
pagecontext[instanceid]= int
```

### Response

**General structure**

```

object {
templates list of (
object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
})
)pluginbaseurl string //Url to the tool_lp plugin folder on this Moodle site
navigation list of (
string //HTML for a navigation item that should be on this page
)pagecontextid int //The page context id
canmanage int //Whether the user manage the templates
}

```

**XML-RPC (PHP structure)**

```

Array
(
[templates] =>
 Array
 (
 [0] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
 [visible] => int
 [contextid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [duedateformatted] => string
 [cohortscount] => int
 [planscount] => int
 [canmanage] => int
 [canread] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
)
)
 [pluginbaseurl] => string
 [navigation] =>
 Array
 (
 [0] => string
)
)
[pagecontextid] => int
[canmanage] => int
)

```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="templates">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canread">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="navigation">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

### REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

## Restricted to logged-in users

Yes

## Callable from AJAX

Yes

### tool\_lp\_data\_for\_user\_competency\_summary

Load a summary of a user competency.

## Arguments

### userid (Required)

Data base record id for the user

#### General structure

```
int //Data base record id for the user
```

#### XML-RPC (PHP structure)

```
[userid] => int
```

#### REST (POST parameters)

```
userid= int
```

### competencyid (Required)

Data base record id for the competency



#### General structure

```
int //Data base record id for the competency
```

#### XML-RPC (PHP structure)

```
[competencyid] => int
```

#### REST (POST parameters)

```
competencyid= int
```

## Response

```

General structure

object {
showrelatedcompetencies int //showrelatedcompetencies
cangrade int //cangrade
competency object {
linkedcourses //linkedcourses
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
)relatedcompetencies //relatedcompetencies
list of (
object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
)competency object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
sortorder int //sortorder
parentid int //parentid
path string //path
ruleoutcome int //ruleoutcome
ruletype string //ruletype
ruleconfig string //ruleconfig
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
competencyframeworkid int //competencyframeworkid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
}
framework object {
shortname string //shortname
idnumber string //idnumber
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //visible
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
contextid int //contextid
taxonomies string //taxonomies
id int //id
}
}
}

```

```
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxononyterm string //taxononyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
)framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showLinks
}
pluginbaseurl string //pluginbaseurl
}
user object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
usercompetency object {
userid int //userid
competencyid int //competencyid
status int //status
reviewerid int //reviewerid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canrequestreview int //canrequestreview
canreview int //canreview
gradename string //gradename
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
issstopreviewallowed int //issstopreviewallowed
issstatusidle int //issstatusidle
issstatusinreview int //issstatusinreview
issstatuswaitingforreview int //issstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
}
```



```
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
usercompetencyplan object {
userid int //userid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
planid int //planid
sortorder int //sortorder
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
usercompetencycourse object {
userid int //userid
courseid int //courseid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
evidence //evidence
list of (
object {
usercompetencyid int //usercompetencyid
contextid int //contextid
action int //action
actionuserid int //actionuserid
descidentifier string //descidentifier
descomponent string //descomponent
desca string //desca
url string //url
grade int //grade
note string //note
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
actionuser object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
description string //description
gradename string //gradename
userdate string //userdate
candelete int //candelete
}
)commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
}
```



```
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
}
```

^

**XML-RPC (PHP structure)**

```

Array
(
 [showrelatedcompetencies] => int
 [cangrade] => int
 [competency] =>
 Array
 (
 [linkedcourses] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
 [relatedcompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)
 [competency] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
)
)
)
)

```



```
[timemodified] => int
[usermodified] => int
)
[framework] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [visible] => int
 [scaleid] => int
 [scaleconfiguration] => string
 [contextid] => int
 [taxonomies] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [competenciescount] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
 [hascourses] => int
 [hasrelatedcompetencies] => int
 [scaleid] => int
 [scaleconfiguration] => string
 [taxonomyterm] => string
 [compath] =>
 Array
 (
 [ancestors] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
)
 [framework] =>
 Array
 (
 [id] => int
 [name] => string
 [first] => int
 [last] => int
 [position] => int
)
 [pluginbaseurl] => string
 [pagecontextid] => int
 [showlinks] => int
)
 [pluginbaseurl] => string
)
[user] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
[usercompetency] =>
 Array
 (
 [userid] => int
 [competencyid] => int
 [status] => int
 [reviewerid] => int
)

```



```
[proficiency] => int
[grade] => int
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[canrequestreview] => int
[canreview] => int
[gradename] => string
[isrequestreviewallowed] => int
[iscancelreviewrequestallowed] => int
[isstartreviewallowed] => int
[istopreviewallowed] => int
[isstatusidle] => int
[isstatusinreview] => int
[isstatuswaitingforreview] => int
[proficiencyname] => string
[reviewer] =>
 Array
 (
 [
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
]
)
 [statusname] => string
 [url] => string
)
[usercompetencyplan] =>
 Array
 (
 [
 [userid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [planid] => int
 [sortorder] => int
 [id] => int
 [timemcreated] => int
 [timemmodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
]
)
[usercompetencycourse] =>
 Array
 (
 [
 [userid] => int
 [courseid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timemcreated] => int
 [timemmodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
]
)
[evidence] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [usercompetencyid] => int
 [contextid] => int
 [action] => int
 [actionuserid] => int
 [descidentifier] => string
 [desccomponent] => string
 [desca] => string
 [url] => string
 [grade] => int
 [note] => string
]
)
]
)

```



```
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[actionuser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [description] => string
 [gradename] => string
 [userdate] => string
 [candelete] => int
)
)
[commentarea] =>
 Array
 (
 [component] => string
 [commentareal] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="showrelatedcompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cangrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competency">
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pdfexportfont">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasprogress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfavourite">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeaccess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showshortname">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
</MULTIPLE>
</KEY>
<KEY name="relatedcompetencies">
<MULTIPLE>
<SINGLE>
<KEY name="shortname">
<VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="sortorder">
<VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
<VALUE>int</VALUE>
</KEY>
<KEY name="path">
<VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
<VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
<VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
<VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
<VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
<VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="competency">
<SINGLE>
<KEY name="shortname">
<VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
<KEY name="description">
<VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
<VALUE>int</VALUE>
</KEY>
<KEY name="sortorder">
<VALUE>int</VALUE>
</KEY>
<KEY name="parentid">
<VALUE>int</VALUE>
</KEY>
<KEY name="path">
<VALUE>string</VALUE>
</KEY>
<KEY name="ruleoutcome">
```

^

```
<VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
```



```
</KEY>
</SINGLE>
</KEY>
<KEY name="hascourses">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="taxononyterm">
 <VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
 <SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="pagecontextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showlinks">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="user">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone2">
 <VALUE>string</VALUE>
</KEY>
<KEY name="department">
 <VALUE>string</VALUE>
</KEY>
<KEY name="institution">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="identity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
<SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstartreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
</SINGLE>
```



```
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencyplan">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
```



```
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencycourse">
<SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
</SINGLE>
</KEY>
<KEY name="evidence">
<MULTIPLE>
 <SINGLE>
 <KEY name="usercompetencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="action">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="actionuserid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="descidentifier">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desccomponent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desca">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
```

^

```
</KEY>
<KEY name="grade">
 <VALUE>int</VALUE>
</KEY>
<KEY name="note">
 <VALUE>string</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="actionuser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="description">
 <VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
 <VALUE>string</VALUE>
</KEY>
<KEY name="candelelete">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
```



```

<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="count">
 <VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
 <VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="template">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

`tool_lp_data_for_user_competency_summary_in_course`

Load a summary of a user competency.

#### Arguments

userid (Required)

Data base record id for the user

##### General structure

```
int //Data base record id for the user
```

##### XML-RPC (PHP structure)

```
[userid] => int
```

##### REST (POST parameters)

```
userid= int
```

competencyid (Required)

Data base record id for the competency

##### General structure

```
int //Data base record id for the competency
```

##### XML-RPC (PHP structure)

```
[competencyid] => int
```

##### REST (POST parameters)

```
competencyid= int
```

courseid (Required)

Data base record id for the course

##### General structure

```
int //Data base record id for the course
```



##### XML-RPC (PHP structure)

```
[courseid] => int
```

##### REST (POST parameters)

```
courseid= int
```

#### Response

```

General structure

object {
 usercompetencysummary object {
 showrelatedcompetencies int //showrelatedcompetencies
 cangrade int //cangrade
 }
 competency object {
 linkedcourses //linkedcourses
 list of (
 object {
 id int //id
 fullname string //fullname
 shortname string //shortname
 idnumber string //idnumber
 summary string //summary
 summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 startdate int //startdate
 enddate int //enddate
 visible int //visible
 showactivitydates int //showactivitydates
 showcompletionconditions int //showcompletionconditions
 pdfexportfont string //pdfexportfont
 fullnamedisplay string //fullnamedisplay
 viewurl string //viewurl
 courseimage string //courseimage
 progress int Optional //progress
 hasprogress int //hasprogress
 isfavourite int //isfavourite
 hidden int //hidden
 timeaccess int Optional //timeaccess
 showshortname int //showshortname
 coursecategory string //coursecategory
 }
)
 }
 relatedcompetencies //relatedcompetencies
 list of (
 object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
)
 competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
 framework object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 visible int //visible
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 contextid int //contextid
 taxonomies string //taxonomies
 }
}

```

```

id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxononyterm string //taxononyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
})
framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showlinks
}
pluginbaseurl string //pluginbaseurl
}
user object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
usercompetency object {
userid int //userid
competencyid int //competencyid
status int //status
reviewerid int //reviewerid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canrequestreview int //canrequestreview
canreview int //canreview
gradename string //gradename
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
issstopreviewallowed int //issstopreviewallowed
isstatusidle int //isstatusidle
isstatusinreview int //isstatusinreview
issstatuswaitingforreview int //issstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
}
}

```



```
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
usercompetencyplan object {
userid int //userid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
planid int //planid
sortorder int //sortorder
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
usercompetencycourse object {
userid int //userid
courseid int //courseid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
evidence //evidence
list of (
object {
usercompetencyid int //usercompetencyid
contextid int //contextid
action int //action
actionuserid int //actionuserid
descidentifier string //descidentifier
desccomponent string //desccomponent
desca string //desca
url string //url
grade int //grade
note string //note
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
actionuser object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
description string //description
gradename string //gradename
userdate string //userdate
candelete int //candelete
}
)commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
}
```



```
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
}
course object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
coursemodules //coursemodules
list of (
object {
id int //id
name string //name
url string Optional //url
iconurl string //iconurl
}
)plans //plans
list of (
object {
name string //name
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
userid int //userid
templateid int //templateid
origtemplateid int //origtemplateid
status int //status
duedate int //duedate
reviewerid int //reviewerid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
statusname string //statusname
isbasedontemplate int //isbasedontemplate
canmanage int //canmanage
canrequestreview int //canrequestreview
canreview int //canreview
canbeedited int //canbeedited
isactive int //isactive
isdraft int //isdraft
iscompleted int //iscompleted
isinreview int //isinreview
iswaitingforreview int //iswaitingforreview
isreopenallowed int //isreopenallowed
iscompleteallowed int //iscompleteallowed
isunlinkallowed int //isunlinkallowed
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
isstartreviewallowed int //isstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isapproveallowed int //isapproveallowed
isunapproveallowed int //isunapproveallowed
duedateformatted string //duedateformatted
commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
}
```



```
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
template object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
)pluginbaseurl string //pluginbaseurl
}
```



**XML-RPC (PHP structure)**

```

Array
(
 [usercompetencysummary] =>
 Array
 (
 [showrelatedcompetencies] => int
 [cangrade] => int
 [competency] =>
 Array
 (
 [linkedcourses] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
 [relatedcompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
 [competency] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
)
)
)
)
)
)
)
)
)

```





```

[competencyid] => int
[status] => int
[reviewerid] => int
[proficiency] => int
[grade] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[canrequestreview] => int
[canreview] => int
[gradename] => string
[isrequestreviewallowed] => int
[iscancelreviewrequestallowed] => int
[isstartreviewallowed] => int
[isstopreviewallowed] => int
[isstatusidle] => int
[isstatusinreview] => int
[isstatuswaitingforreview] => int
[proficiencyname] => string
[reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [statusname] => string
 [url] => string
)
)
[userc competencyplan] =>
 Array
 (
 [userid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [planid] => int
 [sortorder] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
)
[userc competencycourse] =>
 Array
 (
 [userid] => int
 [courseid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
)
[evidence] =>
 Array
 (
 [0] =>
 Array
 (
 [usercompetencyid] => int
 [contextid] => int
 [action] => int
 [actionuserid] => int
 [descidentifier] => string
 [desccomponent] => string
 [desca] => string
)
)

```



```

[url] => string
[grade] => int
[note] => string
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[actionuser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [description] => string
 [gradename] => string
 [userdate] => string
 [candelete] => int
)
)
[commentarea] =>
 Array
 (
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
)
)
[course] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
[coursemodules] =>
 Array
 (
 [0] =>
 Array

```





```
[profileimageurlsmall] => string
)
[template] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
 [visible] => int
 [contextid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [duedateformatted] => string
 [cohortscount] => int
 [planscount] => int
 [canmanage] => int
 [canread] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
 [url] => string
)
)
[pluginbaseurl] => string
)
```

^

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="usercompetencysummary">
 <SINGLE>
 <KEY name="showrelatedcompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cangrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competency">
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pdfexportfont">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasprogress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfavourite">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeaccess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showshortname">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 </MULTIPLE>
 </SINGLE>
 </KEY>
 </MULTIPLE>
 </SINGLE>
 </KEY>
 </SINGLE>
 </RESPONSE>
```



```
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>

```



```
<KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="hascourses">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="taxonomyterm">
 <VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
 <SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
</KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
</KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="pagecontextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showlinks">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="user">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
</KEY>
```



```
<KEY name="email">
 <VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone2">
 <VALUE>string</VALUE>
</KEY>
<KEY name="department">
 <VALUE>string</VALUE>
</KEY>
<KEY name="institution">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="identity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstartreviewallowed">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencyplan">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencycourse">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="evidence">
 <MULTIPLE>
 <SINGLE>
 <KEY name="usercompetencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="action">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="actionuserid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="descidentifier">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desccomponent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desca">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
<KEY name="grade">
 <VALUE>int</VALUE>
</KEY>
<KEY name="note">
 <VALUE>string</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="actionuser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="description">
 <VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
 <VALUE>string</VALUE>
</KEY>
<KEY name="candelete">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```
</KEY>
<KEY name="itemid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="count">
 <VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
 <VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="template">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="course">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
</KEY>
<KEY name="pdfexportfont">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
</KEY>
<KEY name="viewurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="courseimage">
 <VALUE>string</VALUE>
</KEY>
<KEY name="progress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasprogress">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isfavourite">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hidden">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timeaccess">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showshortname">
 <VALUE>int</VALUE>
</KEY>
<KEY name="coursescategory">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="coursemodules">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="iconurl">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="plans">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="templateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="origtemplateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="reviewerid">
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="statusname">
<VALUE>string</VALUE>
</KEY>
<KEY name="isbasedontemplate">
<VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
<VALUE>int</VALUE>
</KEY>
<KEY name="canrequestreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="canbeedited">
<VALUE>int</VALUE>
</KEY>
<KEY name="isactive">
<VALUE>int</VALUE>
</KEY>
<KEY name="isdraft">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscompleted">
<VALUE>int</VALUE>
</KEY>
<KEY name="isinreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="iswaitingforreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="isreopenallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscompleteallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isunlinkallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="issstartreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="issstopreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
<VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
<SINGLE>
<KEY name="component">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
 <VALUE>string</VALUE>
</KEY>
<KEY name="itemid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="count">
 <VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
 <VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="template">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
```

^

```
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="template">
<SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canread">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
</SINGLE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_lp\_data\_for\_user\_competency\_summary\_in\_plan**

Load a summary of a user competency.

**Arguments****competencyid** (Required)

Data base record id for the competency

**General structure**

```
int //Data base record id for the competency
```

**XML-RPC (PHP structure)**

```
[competencyid] => int
```

**REST (POST parameters)**

```
competencyid= int
```

**planid** (Required)

Data base record id for the plan

**General structure**

```
int //Data base record id for the plan
```

**XML-RPC (PHP structure)**

```
[planid] => int
```

**REST (POST parameters)**

```
planid= int
```

**Response**

```

General structure

object {
 usercompetencysummary object {
 showrelatedcompetencies int //showrelatedcompetencies
 cangrade int //cangrade
 competency object {
 linkedcourses //linkedcourses
 list of (
 object {
 id int //id
 fullname string //fullname
 shortname string //shortname
 idnumber string //idnumber
 summary string //summary
 summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 startdate int //startdate
 enddate int //enddate
 visible int //visible
 showactivitydates int //showactivitydates
 showcompletionconditions int //showcompletionconditions
 pdfexportfont string //pdfexportfont
 fullnamedisplay string //fullnamedisplay
 viewurl string //viewurl
 courseimage string //courseimage
 progress int Optional //progress
 hasprogress int //hasprogress
 isfavourite int //isfavourite
 hidden int //hidden
 timeaccess int Optional //timeaccess
 showshortname int //showshortname
 coursecategory string //coursecategory
 }
)relatedcompetencies //relatedcompetencies
 list of (
 object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
)competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
 framework object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN)
 visible int //visible
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 contextid int //contextid
 taxonomies string //taxonomies
 }
 }
 }
}

```

```

id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canmanage int //canmanage
competenciescount int //competenciescount
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
hascourses int //hascourses
hasrelatedcompetencies int //hasrelatedcompetencies
scaleid int //scaleid
scaleconfiguration string //scaleconfiguration
taxononyterm string //taxononyterm
comppath object {
ancestors //ancestors
list of (
object {
id int //id
name string //name
first int //first
last int //last
position int //position
})
framework object {
id int //id
name string //name
first int //first
last int //last
position int //position
}
pluginbaseurl string //pluginbaseurl
pagecontextid int //pagecontextid
showlinks int //showlinks
}
pluginbaseurl string //pluginbaseurl
}
user object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
usercompetency object {
userid int //userid
competencyid int //competencyid
status int //status
reviewerid int //reviewerid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
canrequestreview int //canrequestreview
canreview int //canreview
gradename string //gradename
isrequestreviewallowed int //isrequestreviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
issstopreviewallowed int //issstopreviewallowed
isstatusidle int //isstatusidle
isstatusinreview int //isstatusinreview
issstatuswaitingforreview int //issstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
}
}

```

^

```
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
usercompetencyplan object {
userid int //userid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
planid int //planid
sortorder int //sortorder
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
usercompetencycourse object {
userid int //userid
courseid int //courseid
competencyid int //competencyid
proficiency int //proficiency
grade int //grade
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
gradename string //gradename
proficiencyname string //proficiencyname
}
evidence //evidence
list of (
object {
usercompetencyid int //usercompetencyid
contextid int //contextid
action int //action
actionuserid int //actionuserid
descidentifier string //descidentifier
desccomponent string //desccomponent
desca string //desca
url string //url
grade int //grade
note string //note
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
actionuser object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
description string //description
gradename string //gradename
userdate string //userdate
candelete int //candelete
}
)commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
}
```



```

displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
}
plan object {
name string //name
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
userid int //userid
templateid int //templateid
origtemplateid int //origtemplateid
status int //status
duedate int //duedate
reviewerid int //reviewerid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
statusname string //statusname
isbasedontemplate int //isbasedontemplate
canmanage int //canmanage
canrequestreview int //canrequestreview
canreview int //canreview
canbeedited int //canbeedited
isactive int //isactive
isdraft int //isdraft
iscompleted int //iscompleted
isinreview int //isinreview
iswaitingforreview int //iswaitingforreview
isreopenallowed int //isreopenallowed
iscompleteallowed int //iscompleteallowed
isunlinkallowed int //isunlinkallowed
isrequestviewallowed int //isrequestviewallowed
iscancelreviewrequestallowed int //iscancelreviewrequestallowed
issstartreviewallowed int //issstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isapproveallowed int //isapproveallowed
isunapproveallowed int //isunapproveallowed
duedateformatted string //duedateformatted
commentarea object {
component string //component
commentarea string //commentarea
itemid int //itemid
courseid int //courseid
contextid int //contextid
cid string //cid
autostart int //autostart
canpost int //canpost
canview int //canview
count int //count
collapsediconkey string //collapsediconkey
displaytotalcount int //displaytotalcount
displaycancel int //displaycancel
fullwidth int //fullwidth
linktext string //linktext
notoggle int //notoggle
template string //template
canpostorhascomments int //canpostorhascomments
}
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
template object {
shortname string //shortname
description string //description
descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
duedate int //duedate
visible int //visible

```



```
contextid int //contextid
id int //id
timecreated int //timecreated
timemodified int //timemodified
usermodified int //usermodified
duedateformatted string //duedateformatted
cohortscount int //cohortscount
planscount int //planscount
canmanage int //canmanage
canread int //canread
contextname string //contextname
contextnamenoprefix string //contextnamenoprefix
}
url string //url
}
}
```

^

**XML-RPC (PHP structure)**

```

Array
(
 [
 [usercompetencysummary] =>
 Array
 (
 [
 [showrelatedcompetencies] => int
 [cangrade] => int
 [competency] =>
 Array
 (
 [
 [linkedcourses] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
)
)
)
)
)
)
)
)
)
)
)
 [relatedcompetencies] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)
)
)
)
 [competency] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
)
)
)
)

```





```

[competencyid] => int
[status] => int
[reviewerid] => int
[proficiency] => int
[grade] => int
[id] => int
[timecreated] => int
[timemodified] => int
[usermodified] => int
[canrequestreview] => int
[canreview] => int
[gradename] => string
[isrequestreviewallowed] => int
[iscancelreviewrequestallowed] => int
[isstartreviewallowed] => int
[isstopreviewallowed] => int
[isstatusidle] => int
[isstatusinreview] => int
[isstatuswaitingforreview] => int
[proficiencyname] => string
[reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [statusname] => string
 [url] => string
)
)
[usercompetencyplan] =>
 Array
 (
 [userid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [planid] => int
 [sortorder] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
)
[usercompetencycourse] =>
 Array
 (
 [userid] => int
 [courseid] => int
 [competencyid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [gradename] => string
 [proficiencyname] => string
)
)
[evidence] =>
 Array
 (
 [0] =>
 Array
 (
 [usercompetencyid] => int
 [contextid] => int
 [action] => int
 [actionuserid] => int
 [descidentifier] => string
 [desccomponent] => string
 [desca] => string
)
)

```



```

[url] => string
[grade] => int
[note] => string
[id] => int
[timemcreated] => int
[timemmodified] => int
[usermodified] => int
[actionuser] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [description] => string
 [gradename] => string
 [userdate] => string
 [candelete] => int
)
)
[commentarea] =>
 Array
 (
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
)
[plan] =>
 Array
 (
 [name] => string
 [description] => string
 [descriptionformat] => int
 [userid] => int
 [templateid] => int
 [origtemplateid] => int
 [status] => int
 [duedate] => int
 [reviewerid] => int
 [id] => int
 [timemcreated] => int
 [timemmodified] => int
 [usermodified] => int
 [statusname] => string
 [isbasedontemplate] => int
 [canmanage] => int
 [canrequestreview] => int
 [canreview] => int
 [canbeedited] => int
 [isactive] => int
 [isdraft] => int
 [iscompleted] => int
 [isinreview] => int
 [iswaitingforreview] => int
 [isreopnallowed] => int
 [iscompleteallowed] => int
 [isunlinkallowed] => int
 [isrequestreviewallowed] => int
)
)

```



```
[iscancelreviewrequestallowed] => int
[issstartreviewallowed] => int
[isstopreviewallowed] => int
[isapproveallowed] => int
[isunapproveallowed] => int
[duedateformatted] => string
[commentarea] =>
 Array
 (
 [component] => string
 [commentarea] => string
 [itemid] => int
 [courseid] => int
 [contextid] => int
 [cid] => string
 [autostart] => int
 [canpost] => int
 [canview] => int
 [count] => int
 [collapsediconkey] => string
 [displaytotalcount] => int
 [displaycancel] => int
 [fullwidth] => int
 [linktext] => string
 [notoggle] => int
 [template] => string
 [canpostorhascomments] => int
)
 [reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [template] =>
 Array
 (
 [shortname] => string
 [description] => string
 [descriptionformat] => int
 [duedate] => int
 [visible] => int
 [contextid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [duedateformatted] => string
 [cohortscount] => int
 [planscount] => int
 [canmanage] => int
 [canread] => int
 [contextname] => string
 [contextnamenoprefix] => string
)
 [url] => string
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="usercompetencysummary">
 <SINGLE>
 <KEY name="showrelatedcompetencies">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="cangrade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competency">
 <SINGLE>
 <KEY name="linkedcourses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pdfexportfont">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasprogress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfavourite">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeaccess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showshortname">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 </MULTIPLE>
 </SINGLE>
 </KEY>
 </MULTIPLE>
 </SINGLE>
 </KEY>
 </SINGLE>
 </RESPONSE>
```



```
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="relatedcompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
<KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="ruleoutcome">
 <VALUE>int</VALUE>
</KEY>
<KEY name="ruletype">
 <VALUE>string</VALUE>
</KEY>
<KEY name="ruleconfig">
 <VALUE>string</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="taxonomies">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competenciescount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
```



```
<KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="hascourses">
 <VALUE>int</VALUE>
</KEY>
<KEY name="hasrelatedcompetencies">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
</KEY>
<KEY name="taxonomyterm">
 <VALUE>string</VALUE>
</KEY>
<KEY name="comppath">
 <SINGLE>
 <KEY name="ancestors">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
</KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="framework">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
</KEY>
 <KEY name="first">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="last">
 <VALUE>int</VALUE>
</KEY>
 <KEY name="position">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="pagecontextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="showlinks">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="user">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
</KEY>
 </SINGLE>
</KEY>
```



```
<KEY name="email">
 <VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone2">
 <VALUE>string</VALUE>
</KEY>
<KEY name="department">
 <VALUE>string</VALUE>
</KEY>
<KEY name="institution">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="identity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstartreviewallowed">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencyplan">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="proficiencyname">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetencycourse">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="courseid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="evidence">
 <MULTIPLE>
 <SINGLE>
 <KEY name="usercompetencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="action">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="actionuserid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="descidentifier">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desccomponent">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="desca">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
<KEY name="grade">
 <VALUE>int</VALUE>
</KEY>
<KEY name="note">
 <VALUE>string</VALUE>
</KEY>
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="actionuser">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
<KEY name="description">
 <VALUE>string</VALUE>
</KEY>
<KEY name="gradename">
 <VALUE>string</VALUE>
</KEY>
<KEY name="userdate">
 <VALUE>string</VALUE>
</KEY>
<KEY name="candelete">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</KEY>
```



```
</KEY>
<KEY name="itemid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="count">
 <VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
 <VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="template">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</KEY>
<KEY name="plan">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="templateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="origtemplateid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
```

^

```
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="isbasedontemplate">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canmanage">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canrequestreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canbeedited">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isactive">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isdraft">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscompleted">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isinreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iswaitingforreview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isreopenallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscompleteallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isunlinkallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="isunapproveallowed">
 <VALUE>int</VALUE>
</KEY>
<KEY name="duedateformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="commentarea">
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="commentarea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
```



```
<VALUE>int</VALUE>
</KEY>
<KEY name="courseid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="contextid">
 <VALUE>int</VALUE>
</KEY>
<KEY name="cid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="autostart">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canpost">
 <VALUE>int</VALUE>
</KEY>
<KEY name="canview">
 <VALUE>int</VALUE>
</KEY>
<KEY name="count">
 <VALUE>int</VALUE>
</KEY>
<KEY name="collapsediconkey">
 <VALUE>string</VALUE>
</KEY>
<KEY name="displaytotalcount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="displaycancel">
 <VALUE>int</VALUE>
</KEY>
<KEY name="fullwidth">
 <VALUE>int</VALUE>
</KEY>
<KEY name="linktext">
 <VALUE>string</VALUE>
</KEY>
<KEY name="notoggle">
 <VALUE>int</VALUE>
</KEY>
<KEY name="template">
 <VALUE>string</VALUE>
</KEY>
<KEY name="canpostorhascomments">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
```



```
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
<KEY name="template">
<SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="duedateformatted">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="cohortscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="planscount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canread">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="contextname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="contextnamenoprefix">
 <VALUE>string</VALUE>
 </KEY>
</SINGLE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</RESPONSE>
```



#### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`tool_lp_data_for_user_evidence_list_page`

Load the data for the user evidence list page template

**Arguments**

`userid` (Required)

The user ID

**General structure**

```
int //The user ID
```

**XML-RPC (PHP structure)**

```
[userid] => int
```

**REST (POST parameters)**

```
userid= int
```

**Response**

```

General structure

object {
 canmanage int //Can the current user manage the user's evidence
 userid int //The user ID
 pluginbaseurl string //Url to the tool_lp plugin folder on this Moodle site
 evidence list of (
 object {
 userid int //userid
 name string //name
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 url string //url
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 canmanage int //canmanage
 filecount int //filecount
 files //files
 list of (
 object {
 contextid int //contextid
 component string //component
 filearea string //filearea
 itemid int //itemid
 filepath string //filepath
 filename string //filename
 isdir int //isdir
 isimage int //isimage
 timemodified int //timemodified
 timecreated int //timecreated
 filesize int //filesize
 author string //author
 license string //license
 filenameshort string //filenameshort
 filesizeformatted string //filesizeformatted
 icon string //icon
 timecreatedformatted string //timecreatedformatted
 timemodifiedformatted string //timemodifiedformatted
 url string //url
 }
)hasurlorfiles int //hasurlorfiles
 urlshort string //urlshort
 competencycount int //competencycount
 usercompetencies Optional //usercompetencies
 list of (
 object {
 competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 rulename string //rulename
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
 usercompetency object {
 userid int //userid
 competencyid int //competencyid
 status int //status
 reviewerid int //reviewerid
 proficiency int //proficiency
 grade int //grade
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 canrequestreview int //canrequestreview
 canreview int //canreview
 gradename string //gradename
 isrequestreviewallowed int //isrequestreviewallowed
 iscancelreviewrequestallowed int //iscancelreviewrequestallowed
 }
 }
)
 }
)
}

```



```
isstartreviewallowed int //isstartreviewallowed
isstopreviewallowed int //isstopreviewallowed
isstatusidle int //isstatusidle
isstatusinreview int //isstatusinreview
isstatuswaitingforreview int //isstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
 id int //id
 email string //email
 idnumber string //idnumber
 phone1 string //phone1
 phone2 string //phone2
 department string //department
 institution string //institution
 fullname string //fullname
 identity string //identity
 profileurl string //profileurl
 profileimageurl string //profileimageurl
 profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
)
)userhasplan int //userhasPlan
)
)navigation list of (
string //HTML for a navigation item that should be on this page
)}
```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [canmanage] => int
 [userid] => int
 [pluginbaseurl] => string
 [evidence] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [userid] => int
 [name] => string
 [description] => string
 [descriptionformat] => int
 [url] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [filecount] => int
 [files] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [contextid] => int
 [component] => string
 [filearea] => string
 [itemid] => int
 [filepath] => string
 [filename] => string
 [isdir] => int
 [isimage] => int
 [timemodified] => int
 [timecreated] => int
 [filesize] => int
 [author] => string
 [license] => string
 [filenameshort] => string
 [filesizeformatted] => string
 [icon] => string
 [timecreatedformatted] => string
 [timemodifiedformatted] => string
 [url] => string
)
)
)
)
)
)
 [hasurlorfiles] => int
 [urlshort] => string
 [competencycount] => int
 [usercompetencies] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [competency] =>
 Array
 (
 [
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
)
)
)
)
)
 [usercompetency] =>
)
)
)
)
)
)
)
)
)
)

```

^

```

 Array
 (
 [userid] => int
 [competencyid] => int
 [status] => int
 [reviewerid] => int
 [proficiency] => int
 [grade] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canrequestreview] => int
 [canreview] => int
 [gradename] => string
 [isrequestreviewallowed] => int
 [iscancelreviewrequestallowed] => int
 [isstartreviewallowed] => int
 [isstopreviewallowed] => int
 [isstatusidle] => int
 [isstatusinreview] => int
 [isstatuswaitingforreview] => int
 [proficiencyname] => string
 [reviewer] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
 [statusname] => string
 [url] => string
)
)
)
)
[navigation] =>
 Array
 (
 [0] => string
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="evidence">
 <MULTIPLE>
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filecount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filearea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isdir">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isimage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



```
<KEY name="filesize">
 <VALUE>int</VALUE>
</KEY>
<KEY name="author">
 <VALUE>string</VALUE>
</KEY>
<KEY name="license">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filenameshort">
 <VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="hasurlorfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="urlshort">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencycount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usercompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
```



```
<KEY name="id">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
 <VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="competencyid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="status">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reviewerid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiency">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="grade">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canrequestreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="gradename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isrequestreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="iscancelreviewrequestallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstartreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstopreviewallowed">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstatusidle">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstatusinreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isstatuswaitingforreview">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="proficiencyname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="reviewer">
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>

```



```

</KEY>
<KEY name="email">
 <VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
 <VALUE>string</VALUE>
</KEY>
<KEY name="phone2">
 <VALUE>string</VALUE>
</KEY>
<KEY name="department">
 <VALUE>string</VALUE>
</KEY>
<KEY name="institution">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="identity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</MULTIPLE>
</KEY>
<KEY name="userhasplan">
 <VALUE>int</VALUE>
</KEY>
<SINGLE>
</MULTIPLE>
</KEY>
<KEY name="navigation">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

```

REST

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

**Callable from AJAX**

Yes

**tool\_lp\_data\_for\_user\_evidence\_page**

Load the data for the user evidence page template

**Arguments**

id (Required)

The user evidence ID

**General structure**

```
int //The user evidence ID
```

**XML-RPC (PHP structure)**

```
[id] => int
```

**REST (POST parameters)**

```
id= int
```

**Response**

```

General structure

object {
 userevidence object {
 userid int //userid
 name string //name
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 url string //url
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 canmanage int //canmanage
 filecount int //filecount
 files //files
 }
 list of (
 object {
 contextid int //contextid
 component string //component
 filearea string //filearea
 itemid int //itemid
 filepath string //filepath
 filename string //filename
 isdir int //isdir
 isimage int //isimage
 timemodified int //timemodified
 timecreated int //timecreated
 filesize int //filesize
 author string //author
 license string //license
 filenameshort string //filenameshort
 filesizeformatted string //filesizeformatted
 icon string //icon
 timecreatedformatted string //timecreatedformatted
 timemodifiedformatted string //timemodifiedformatted
 url string //url
 }
)
)hasurlorfiles int //hasurlorfiles
 urlshort string //urlshort
 competencycount int //competencycount
 usercompetencies Optional //usercompetencies
 list of (
 object {
 competency object {
 shortname string //shortname
 idnumber string //idnumber
 description string //description
 descriptionformat int Default to "1" //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
 sortorder int //sortorder
 parentid int //parentid
 path string //path
 ruleoutcome int //ruleoutcome
 ruletype string //ruletype
 ruleconfig string //ruleconfig
 scaleid int //scaleid
 scaleconfiguration string //scaleconfiguration
 competencyframeworkid int //competencyframeworkid
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 }
 usercompetency object {
 userid int //userid
 competencyid int //competencyid
 status int //status
 reviewerid int //reviewerid
 proficiency int //proficiency
 grade int //grade
 id int //id
 timecreated int //timecreated
 timemodified int //timemodified
 usermodified int //usermodified
 canrequestreview int //canrequestreview
 canreview int //canreview
 gradename string //gradename
 isrequestreviewallowed int //isrequestreviewallowed
 iscancelreviewrequestallowed int //iscancelreviewrequestallowed
 issstartreviewallowed int //issstartreviewallowed
 isstopreviewallowed int //isstopreviewallowed
 isstatusidle int //isstatusidle
 isstatusinreview int //isstatusinreview
 }
 }
)
}

```



```
isstatuswaitingforreview int //isstatuswaitingforreview
proficiencyname string //proficiencyname
reviewer object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
statusname string //statusname
url string //url
}
)
)userhasplan int //userhasPlan
)
pluginbaseurl string //Url to the tool_lp plugin folder on this Moodle site
}
```

^

**XML-RPC (PHP structure)**

```

Array
(
 [userevidence] =>
 Array
 (
 [userid] => int
 [name] => string
 [description] => string
 [descriptionformat] => int
 [url] => string
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
 [canmanage] => int
 [filecount] => int
 [files] =>
 Array
 (
 [0] =>
 Array
 (
 [contextid] => int
 [component] => string
 [filearea] => string
 [itemid] => int
 [filepath] => string
 [filename] => string
 [isdir] => int
 [isimage] => int
 [timemodified] => int
 [timecreated] => int
 [filesize] => int
 [author] => string
 [license] => string
 [filenameshort] => string
 [filesizeformatted] => string
 [icon] => string
 [timecreatedformatted] => string
 [timemodifiedformatted] => string
 [url] => string
)
)
 [hasurlorfiles] => int
 [urlshort] => string
 [competencycount] => int
 [usercompetencies] =>
 Array
 (
 [0] =>
 Array
 (
 [competency] =>
 Array
 (
 [shortname] => string
 [idnumber] => string
 [description] => string
 [descriptionformat] => int
 [sortorder] => int
 [parentid] => int
 [path] => string
 [ruleoutcome] => int
 [ruletype] => string
 [ruleconfig] => string
 [scaleid] => int
 [scaleconfiguration] => string
 [competencyframeworkid] => int
 [id] => int
 [timecreated] => int
 [timemodified] => int
 [usermodified] => int
)
 [usercompetency] =>
 Array
 (
 [userid] => int
 [competencyid] => int
 [status] => int
 [reviewerid] => int
)
)
)
)
)
)

```





## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="userevidence">
 <SINGLE>
 <KEY name="userid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="canmanage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filecount">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="contextid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filearea">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isdir">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isimage">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="author">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="license">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filenameshort">
```



```
<VALUE>string</VALUE>
</KEY>
<KEY name="filesizeformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="icon">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timecreatedformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="timemodifiedformatted">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="hasurlorfiles">
 <VALUE>int</VALUE>
</KEY>
<KEY name="urlshort">
 <VALUE>string</VALUE>
</KEY>
<KEY name="competencycount">
 <VALUE>int</VALUE>
</KEY>
<KEY name="usercompetencies">
 <MULTIPLE>
 <SINGLE>
 <KEY name="competency">
 <SINGLE>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="sortorder">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="parentid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="path">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleoutcome">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="ruletype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="ruleconfig">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="scaleid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="scaleconfiguration">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="competencyframeworkid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timecreated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="usermodified">
```



```
<VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="usercompetency">
<SINGLE>
<KEY name="userid">
<VALUE>int</VALUE>
</KEY>
<KEY name="competencyid">
<VALUE>int</VALUE>
</KEY>
<KEY name="status">
<VALUE>int</VALUE>
</KEY>
<KEY name="reviewerid">
<VALUE>int</VALUE>
</KEY>
<KEY name="proficiency">
<VALUE>int</VALUE>
</KEY>
<KEY name="grade">
<VALUE>int</VALUE>
</KEY>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="timecreated">
<VALUE>int</VALUE>
</KEY>
<KEY name="timemodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="usermodified">
<VALUE>int</VALUE>
</KEY>
<KEY name="canrequestreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="canreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="gradename">
<VALUE>string</VALUE>
</KEY>
<KEY name="isrequestreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="iscancelreviewrequestallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstartreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="istopreviewallowed">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstatusidle">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstatusinreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="isstatuswaitingforreview">
<VALUE>int</VALUE>
</KEY>
<KEY name="proficiencyname">
<VALUE>string</VALUE>
</KEY>
<KEY name="reviewer">
<SINGLE>
<KEY name="id">
<VALUE>int</VALUE>
</KEY>
<KEY name="email">
<VALUE>string</VALUE>
</KEY>
<KEY name="idnumber">
<VALUE>string</VALUE>
</KEY>
<KEY name="phone1">
<VALUE>string</VALUE>
</KEY>
```

^

```

<KEY name="phone2">
 <VALUE>string</VALUE>
</KEY>
<KEY name="department">
 <VALUE>string</VALUE>
</KEY>
<KEY name="institution">
 <VALUE>string</VALUE>
</KEY>
<KEY name="fullname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="identity">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurl">
 <VALUE>string</VALUE>
</KEY>
<KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="statusname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="url">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</SINGLE>
</MULTIPLE>
</KEY>
<KEY name="userhasplan">
 <VALUE>int</VALUE>
</KEY>
</SINGLE>
</KEY>
<KEY name="pluginbaseurl">
 <VALUE>string</VALUE>
</KEY>
</SINGLE>
</KEY>
</RESPONSE>

```

### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

### tool\_lp\_list\_courses\_using\_competency

List the courses using a competency

### Arguments

**id** (Required)

The competency id

#### General structure

```
int //The competency id
```

#### XML-RPC (PHP structure)

```
[id] => int
```

#### REST (POST parameters)

```
id= int
```

### Response

#### General structure

```
list of (
object {
id int //id
fullname string //fullname
shortname string //shortname
idnumber string //idnumber
summary string //summary
summaryformat int //summary format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
startdate int //startdate
enddate int //enddate
visible int //visible
showactivitydates int //showactivitydates
showcompletionconditions int //showcompletionconditions
pdfexportfont string //pdfexportfont
fullnamedisplay string //fullnamedisplay
viewurl string //viewurl
courseimage string //courseimage
progress int Optional //progress
hasprogress int //hasprogress
isfavourite int //isfavourite
hidden int //hidden
timeaccess int Optional //timeaccess
showshortname int //showshortname
coursecategory string //coursecategory
}
)
```

**XML-RPC (PHP structure)**

```
Array
(
 [0] =>
 Array
 (
 [id] => int
 [fullname] => string
 [shortname] => string
 [idnumber] => string
 [summary] => string
 [summaryformat] => int
 [startdate] => int
 [enddate] => int
 [visible] => int
 [showactivitydates] => int
 [showcompletionconditions] => int
 [pdfexportfont] => string
 [fullnamedisplay] => string
 [viewurl] => string
 [courseimage] => string
 [progress] => int
 [hasprogress] => int
 [isfavourite] => int
 [hidden] => int
 [timeaccess] => int
 [showshortname] => int
 [coursecategory] => string
)
)
)
```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summary">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="summaryformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="startdate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enddate">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showactivitydates">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showcompletionconditions">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="pdfexportfont">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullnamedisplay">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="progress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hasprogress">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="isfavourite">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="timeaccess">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="showshortname">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</RESPONSE>
```



## Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_lp\_search\_cohorts**

Search for cohorts. This method is deprecated, please call 'core\_cohort\_search\_cohorts' instead

**Arguments****query** (Required)

Query string

**General structure**

```
string //Query string
```

**XML-RPC (PHP structure)**

```
[query] => string
```

**REST (POST parameters)**

```
query= string
```

**context** (Required)**General structure**

```
object {
 contextid int Default to "0" //Context ID. Either use this value, or Level and instanceid.
 contextlevel string Default to "" //Context Level. To be used with instanceid.
 instanceid int Default to "0" //Context instance ID. To be used with level
}
```

**XML-RPC (PHP structure)**

```
[context] =>
 Array
 (
 [contextid] => int
 [contextlevel] => string
 [instanceid] => int
)
```

**REST (POST parameters)**

```
context[contextid]= int
context[contextlevel]= string
context[instanceid]= int
```

includes (Default to "parents")

What other contexts to fetch the frameworks from. (all, parents, self)

**General structure**

```
string Default to "parents" //What other contexts to fetch the frameworks from. (all, parents, self)
```

**XML-RPC (PHP structure)**

```
[includes] => string
```

**REST (POST parameters)**

```
includes= string
```

limitfrom (Default to "0")

limitfrom we are fetching the records from

**General structure**

```
int Default to "0" //limitfrom we are fetching the records from
```

**XML-RPC (PHP structure)**

```
[limitfrom] => int
```

**REST (POST parameters)**

```
limitfrom= int
```

limitnum (Default to "25")

Number of records to fetch

**General structure**

```
int Default to "25" //Number of records to fetch
```

**XML-RPC (PHP structure)**

```
[limitnum] => int
```

**REST (POST parameters)**

```
limitnum= int
```

**Response**

**General structure**

```

object {
cohorts list of (
object {
id int //ID of the cohort
name string //cohort name
idnumber string //cohort idnumber
description string //cohort description
descriptionformat int //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN, or 4 = MARKDOWN
visible int //cohort visible
theme string Optional //cohort theme
customfields Optional //Custom fields
list of (
object {
name string //The name of the custom field
shortname string //The shortname of the custom field - to be able to build the field class in the code
type string //The type of the custom field - text field, checkbox...
valueraw string //The raw value of the custom field
value string //The value of the custom field
})
)
)
}

```

**XML-RPC (PHP structure)**

```

Array
(
[cohorts] =>
 Array
 (
 [
0] =>
 Array
 (
 [
id] => int
 [
name] => string
 [
idnumber] => string
 [
description] => string
 [
descriptionformat] => int
 [
visible] => int
 [
theme] => string
 [
customfields] =>
 Array
 (
 [
0] =>
 Array
 (
 [
name] => string
 [
shortname] => string
 [
type] => string
 [
valueraw] => string
 [
value] => string
)
)
)
)
)
)

```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="cohorts">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="description">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="descriptionformat">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="visible">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="theme">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="customfields">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="shortname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="type">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="valueraw">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**tool\_lp\_search\_users**

Search for users.

**Arguments**

**query** (Required)

Query string

**General structure**

```
string //Query string
```

**XML-RPC (PHP structure)**

```
[query] => string
```

**REST (POST parameters)**

```
query= string
```

**capability** (Required)

Required capability

**General structure**

```
string //Required capability
```

**XML-RPC (PHP structure)**

```
[capability] => string
```

**REST (POST parameters)**

```
capability= string
```

**limitfrom** (Default to "0")

Number of records to skip

**General structure**

```
int Default to "0" //Number of records to skip
```

**XML-RPC (PHP structure)**

```
[limitfrom] => int
```

**REST (POST parameters)**

```
limitfrom= int
```

**limitnum** (Default to "100")

Number of records to fetch

**General structure**

```
string Default to "100" //Number of records to fetch
```

**XML-RPC (PHP structure)**

```
[limitnum] => string
```

**REST (POST parameters)**

```
limitnum= string
```

**Response****General structure**

```
object {
users list of (
object {
id int //id
email string //email
idnumber string //idnumber
phone1 string //phone1
phone2 string //phone2
department string //department
institution string //institution
fullname string //fullname
identity string //identity
profileurl string //profileurl
profileimageurl string //profileimageurl
profileimageurlsmall string //profileimageurlsmall
}
)count int //Total number of results.
}
```

**XML-RPC (PHP structure)**

```
Array
(
[users] =>
 Array
 (
 [0] =>
 Array
 (
 [id] => int
 [email] => string
 [idnumber] => string
 [phone1] => string
 [phone2] => string
 [department] => string
 [institution] => string
 [fullname] => string
 [identity] => string
 [profileurl] => string
 [profileimageurl] => string
 [profileimageurlsmall] => string
)
)
 [count] => int
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="users">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="email">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="idnumber">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone1">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="phone2">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="department">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="institution">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identity">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="profileimageurlsmall">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="count">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_mobile\_call\_external\_functions**

Call multiple external functions and return all responses.

**Arguments**

**requests** (Required)

**General structure**

```
list of (
object {
function string //Function name
arguments string Default to "{}" //JSON-encoded object with named arguments
settingraw int Default to "" //Return raw text
settingfilter int Default to "" //Filter text
settingfileurl int Default to "1" //Rewrite plugin file URLs
settinglang string Default to "" //Session Language
}
)
```

**XML-RPC (PHP structure)**

```
[requests] =>
Array
(
[0] =>
Array
(
[function] => string
[arguments] => string
[settingraw] => int
[settingfilter] => int
[settingfileurl] => int
[settinglang] => string
)
)
```

**REST (POST parameters)**

```
requests[0][function]= string
requests[0][arguments]= string
requests[0][settingraw]= int
requests[0][settingfilter]= int
requests[0][settingfileurl]= int
requests[0][settinglang]= string
```

**Response****General structure**

```
object {
responses list of (
object {
error int //Whether an exception was thrown.
data string Optional //JSON-encoded response data
exception string Optional //JSON-encoded exception info
}
)}
```

## XML-RPC (PHP structure)

```
Array
 (
 [responses] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [error] => int
 [data] => string
 [exception] => string
)
)
)
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="responses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="error">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="data">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="exception">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

## Callable from AJAX

No

tool\_mobile\_get\_autologin\_key

Creates an auto-login key for the current user. Is created only in https sites and is restricted by time, ip address and only works if the request comes from the Moodle mobile or desktop app.

## Arguments

**privatetoken** (Required)

Private token, usually generated by login/token.php

**General structure**

```
string //Private token, usually generated by Login/token.php
```

**XML-RPC (PHP structure)**

```
[privatetoken] => string
```

**REST (POST parameters)**

```
privatetoken= string
```

**Response****General structure**

```
object {
key string //Auto-Login key for a single usage with time expiration.
autologinurl string //Auto-Login URL.
warnings Optional //List of warnings
list of (
 //warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
}
)}
```

**XML-RPC (PHP structure)**

```
Array
(
[key] => string
[autologinurl] => string
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="key">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="autologinurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**tool\_mobile\_get\_config**

Returns a list of the site configurations, filtering by section.

**Arguments****section** (Default to "")

Settings section name.

**General structure**

```
string Default to "" //Settings section name.
```

**XML-RPC (PHP structure)**

```
[section] => string
```

**REST (POST parameters)**

```
section= string
```

**Response**

**General structure**

```
object {
 settings //Settings
 list of (
 object {
 name string //The name of the setting
 value string //The value of the setting
 }
)warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [settings] =>
 Array
 (
 [0] =>
 Array
 (
 [name] => string
 [value] => string
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="settings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

No

**tool\_mobile\_get\_content**

Returns a piece of content to be displayed in the Mobile app.

**Arguments**

component (Required)

Component where the class is e.g. mod\_assign.

**General structure**

```
string //Component where the class is e.g. mod_assign.
```

**XML-RPC (PHP structure)**

```
[component] => string
```

**REST (POST parameters)**

```
component= string
```

**method** (Required)

Method to execute in class \\${component}\output\mobile.

**General structure**

```
string //Method to execute in class \${component}\output\mobile.
```

**XML-RPC (PHP structure)**

```
[method] => string
```

**REST (POST parameters)**

```
method= string
```

**args** (Optional)

Args for the method are optional.

**General structure**

*Optional //Args for the method are optional.*

```
list of (
object {
name string //Param name.
value string //Param value.
}
)
```

**XML-RPC (PHP structure)**

```
[args] =>
Array
(
[0] =>
Array
(
[name] => string
[value] => string
)
)
```

**REST (POST parameters)**

```
args[0][name]= string
args[0][value]= string
```

**Response**

**General structure**

```
object {
templates //Templates required by the generated content.
list of (
object {
id string //ID of the template.
html string //HTML code.
})
javascript string //JavaScript code.
otherdata //Other data that can be used or manipulated by the template via 2-way data-binding.
list of (
object {
name string //Field name.
value string //Field value.
})
files //Files in the content.
list of (
//File.
object {
filename string Optional //File name.
filepath string Optional //File path.
filesize int Optional //File size.
fileurl string Optional //Downloadable file url.
timemodified int Optional //Time modified.
mimetype string Optional //File mime type.
isexternalfile int Optional //Whether is an external file.
repositorytype string Optional //The repository type for external files.
})
restrict //Restrict this content to certain users or courses.
object {
users Optional //List of allowed users.
list of (
int //user id
)courses Optional //List of allowed courses.
list of (
int //course id
)
disabled int Optional //Whether we consider this disabled or not.
})
```



**XML-RPC (PHP structure)**

```
Array
(
 [
 [templates] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [id] => string
 [html] => string
)
)
)
 [javascript] => string
 [otherdata] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [value] => string
)
)
)
 [files] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [filename] => string
 [filepath] => string
 [filesize] => int
 [fileurl] => string
 [timemodified] => int
 [mimetype] => string
 [isexternalfile] => int
 [repositorytype] => string
)
)
)
 [restrict] =>
 Array
 (
 [
 [users] =>
 Array
 (
 [
 [0] => int
)
)
 [courses] =>
 Array
 (
 [
 [0] => int
)
)
)
)
 [disabled] => int
)
)
```



## REST

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="templates">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="html">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="javascript">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="otherdata">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="value">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="files">
 <MULTIPLE>
 <SINGLE>
 <KEY name="filename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filepath">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="timemodified">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="mimetype">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="isexternalfile">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="repositorytype">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="restrict">
 <SINGLE>
 <KEY name="users">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="courses">
 <MULTIPLE>
 <VALUE>int</VALUE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="disabled">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>

```



&lt;/RESPONSE&gt;

### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

### Restricted to logged-in users

Yes

### Callable from AJAX

No

### tool\_mobile\_get\_plugins\_supporting\_mobile

Returns a list of Moodle plugins supporting the mobile app.

### Arguments

### Response

**General structure**

```
object {
 plugins list of (
 object {
 component string //The plugin component name.
 version string //The plugin version number.
 addon string //The Mobile addon (package) name.
 dependencies //The list of Mobile addons this addon depends on.
 list of (
 string //Mobile addon name.
)fileurl string //The addon package url for download
 or empty if it doesn't exist.
 filehash string //The addon package hash or empty if it doesn't exist.
 filesize int //The addon package size or empty if it doesn't exist.
 handlers string Optional //Handlers definition (JSON)
 lang string Optional //Language strings used by the handlers (JSON)
 }
)warnings Optional //List of warnings
 list of (
 //warning
)object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [plugins] =>
 Array
 (
 [0] =>
 Array
 (
 [component] => string
 [version] => string
 [addon] => string
 [dependencies] =>
 Array
 (
 [0] => string
)
 [fileurl] => string
 [filehash] => string
 [filesize] => int
 [handlers] => string
 [lang] => string
)
)
)
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
```



REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="plugins">
 <MULTIPLE>
 <SINGLE>
 <KEY name="component">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="version">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="addon">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="dependencies">
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
 </KEY>
 <KEY name="fileurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filehash">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="filesize">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="handlers">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="lang">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

**Callable from AJAX**

Yes

**tool\_mobile\_get\_public\_config**

Returns a list of the site public settings, those not requiring authentication.

**Arguments****Response****General structure**

```

object {
 wwwroot string //Site URL.
 httpswwwroot string //Site https URL (if httpsLogin is enabled).
 sitename string //Site name.
 guestlogin int //Whether guest login is enabled.
 rememberusername int //Values: 0 for No, 1 for Yes, 2 for optional.
 authloginviaemail int //Whether log in via email is enabled.
 registerauth string //Authentication method for user registration.
 forgottenpasswordurl string //Forgotten password URL.
 authinstructions string //Authentication instructions.
 authnoneenabled int //Whether auth none is enabled.
 enablewebservices int //Whether Web Services are enabled.
 enablemobilewebservice int //Whether the Mobile service is enabled.
 maintenanceenabled int //Whether site maintenance is enabled.
 maintenancemessage string //Maintenance message.
 logourl string Optional //The site logo URL.
 compactlogourl string Optional //The site compact logo URL.
 typeoflogin int //The type of login. 1 for app, 2 for browser, 3 for embedded.
 launchurl string Optional //SSO Login Launch URL.
 mobilecssurl string Optional //Mobile custom CSS theme.
 tool_mobile_disabledfeatures string Optional //Disabled features in the app.
 identityproviders Optional //Identity providers
 list of (
 object {
 name string //The identity provider name.
 iconurl string //The icon URL for the provider.
 url string //The URL of the provider.
 }
)country string Optional //Default site country
 agedigitalconsentverification int Optional //Whether age digital consent verification
 is enabled.
 supportname string Optional //Site support contact name
 (only if age verification is enabled).
 supportemail string Optional //Site support contact email
 (only if age verification is enabled).
 supportpage string Optional //Site support page link.
 supportavailability int Optional //Determines who has access to contact site support.
 autolang int Optional //Whether to detect default language
 from browser setting.
 lang string Optional //Default language for the site.
 langmenu int Optional //Whether the language menu should be displayed.
 langlist string Optional //Languages on language menu.
 locale string Optional //Sitewide locale.
 tool_mobile_minimumversion string Optional //Minimum required version to access.
 tool_mobile_iosappid string Optional //iOS app's unique identifier.
 tool_mobile_androidappid string Optional //Android app's unique identifier.
 tool_mobile_setuplink string Optional //App download page.
 tool_mobile_qrcodetype int Optional //QR login configuration.
 warnings Optional //List of warnings
 list of (
 /warning
)object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated English message to explain the warning
 }
})
}

```



**XML-RPC (PHP structure)**

```

Array
(
 [
 [wwwroot] => string
 [httpswwwroot] => string
 [sitename] => string
 [guestlogin] => int
 [rememberusername] => int
 [authloginviaemail] => int
 [registerauth] => string
 [forgottenpasswordurl] => string
 [authinstructions] => string
 [authnoneenabled] => int
 [enablewebservices] => int
 [enablemobilewebservice] => int
 [maintenanceenabled] => int
 [maintenancemessage] => string
 [logourl] => string
 [compactlogourl] => string
 [typeoflogin] => int
 [launchurl] => string
 [mobilecssurl] => string
 [tool_mobile_disabledfeatures] => string
 [identityproviders] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [name] => string
 [iconurl] => string
 [url] => string
)
)
)
)
 [country] => string
 [agedigitalconsentverification] => int
 [supportname] => string
 [supportemail] => string
 [supportpage] => string
 [supportavailability] => int
 [autolang] => int
 [lang] => string
 [langmenu] => int
 [langlist] => string
 [locale] => string
 [tool_mobile_minimumversion] => string
 [tool_mobile_iosappid] => string
 [tool_mobile_androidappid] => string
 [tool_mobile_setuplink] => string
 [tool_mobile_qrcodetype] => int
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```



## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="wwwroot">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="httpswwwroot">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="sitename">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="guestlogin">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="rememberusername">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="authloginviaemail">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="registerauth">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="forgottenpasswordurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authinstructions">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="authnoneenabled">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enablewebservices">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="enablenewservice">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maintenanceenabled">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="maintenancemessage">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="logourl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="compactlogourl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="typeoflogin">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="launchurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="mobilecssurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="tool_mobile_disabledfeatures">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="identityproviders">
 <MULTIPLE>
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="iconurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="url">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="country">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
```



```

<KEY name="agedigitalconsentverification">
 <VALUE>int</VALUE>
</KEY>
<KEY name="supportname">
 <VALUE>string</VALUE>
</KEY>
<KEY name="supportemail">
 <VALUE>string</VALUE>
</KEY>
<KEY name="supportpage">
 <VALUE>string</VALUE>
</KEY>
<KEY name="supportavailability">
 <VALUE>int</VALUE>
</KEY>
<KEY name="autolang">
 <VALUE>int</VALUE>
</KEY>
<KEY name="lang">
 <VALUE>string</VALUE>
</KEY>
<KEY name="langmenu">
 <VALUE>int</VALUE>
</KEY>
<KEY name="langlist">
 <VALUE>string</VALUE>
</KEY>
<KEY name="locale">
 <VALUE>string</VALUE>
</KEY>
<KEY name="tool_mobile_minimumversion">
 <VALUE>string</VALUE>
</KEY>
<KEY name="tool_mobile_iosappid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="tool_mobile_androidappid">
 <VALUE>string</VALUE>
</KEY>
<KEY name="tool_mobile_setuplink">
 <VALUE>string</VALUE>
</KEY>
<KEY name="tool_mobile_qrcodetype">
 <VALUE>int</VALUE>
</KEY>
<KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
</KEY>
</SINGLE>
</RESPONSE>

```



### Error message

#### REST

```

<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>

```

Restricted to logged-in users

No

Callable from AJAX

Yes

### tool\_mobile\_get\_tokens\_for\_qr\_login

Returns a WebService token (and private token) for QR login.

#### Arguments

**qrloginkey** (Required)

The user key for validating the request.

##### General structure

```
string //The user key for validating the request.
```

##### XML-RPC (PHP structure)

```
[qrloginkey] => string
```

##### REST (POST parameters)

```
qrloginkey= string
```

**userid** (Required)

The user the key belongs to.

##### General structure

```
int //The user the key belongs to.
```

##### XML-RPC (PHP structure)

```
[userid] => int
```

##### REST (POST parameters)

```
userid= int
```



#### Response

##### General structure

```
object {
 token string //A valid WebService token for the official mobile app service.
 privatetoken string //Private token used for auto-Login processes.
 warnings Optional //list of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)}
```

## XML-RPC (PHP structure)

```
Array
(
 [token] => string
 [privatetoken] => string
 [warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="token">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="privatetoken">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```



## Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

**tool\_mobile\_validate\_subscription\_key**

Check if the given site subscription key is valid.

**Arguments****key** (Required)

Site subscription temporary key.

**General structure**

```
string //Site subscription temporary key.
```

**XML-RPC (PHP structure)**

```
[key] => string
```

**REST (POST parameters)**

```
key= string
```

**Response****General structure**

```
object {
validated int //Whether the key is validated or not.
warnings Optional //List of warnings
list of (
//warning
object {
item string Optional //item
itemid int Optional //item id
warningcode string //the warning code can be used by the client app to implement specific behaviour
message string //untranslated english message to explain the warning
})
})}
```

**XML-RPC (PHP structure)**

```
Array
(
[validated] => int
[warnings] =>
 Array
 (
 [0] =>
 Array
 (
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="validated">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

**Callable from AJAX**

Yes

**tool\_moodlenet\_search\_courses**

For some given input search for a course that matches

**Arguments****searchvalue** (Required)

search value

**General structure**

string //search value

**XML-RPC (PHP structure)**

[searchvalue] =&gt; string

## REST (POST parameters)

```
searchvalue= string
```

## Response

## General structure

```
object {
courses list of (
object {
id int //course id
fullname string //course full name
hidden int //is the course visible
viewurl string //Next step of import
coursecategory string //Category name
courseimage string //course image
}
)}
```

## XML-RPC (PHP structure)

```
Array
(
[courses] =>
 Array
 (
 [
0] =>
 Array
 (
 [
id] => int
 [
fullname] => string
 [
hidden] => int
 [
viewurl] => string
 [
coursecategory] => string
 [
courseimage] => string
)
)
)
```

## REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="courses">
 <MULTIPLE>
 <SINGLE>
 <KEY name="id">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="fullname">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="hidden">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="viewurl">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="coursecategory">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="courseimage">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_moodlenet\_verify\_webfinger**

Verify if the passed information resolves into a WebFinger profile URL

**Arguments****profileurl (Required)**

The profile url that the user has given us

**General structure**

```
string //The profile url that the user has given us
```

**XML-RPC (PHP structure)**

```
[profileurl] => string
```

**REST (POST parameters)**

```
profileurl= string
```

**course (Required)**

The course we are adding to

**General structure**

```
int //The course we are adding to
```

**XML-RPC (PHP structure)**

```
[course] => int
```

**REST (POST parameters)**

```
course= int
```

**section (Required)**

The section within the course we are adding to

**General structure**

```
int //The section within the course we are adding to
```

**XML-RPC (PHP structure)**

```
[section] => int
```

**REST (POST parameters)**

```
section= int
```

**Response****General structure**

```
object {
 result int //Was the passed content a valid WebFinger?
 message string //Our message for the user
 domain string Optional //Domain to redirect the user to
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [result] => int
 [message] => string
 [domain] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="domain">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_policy\_get\_policy\_version**

Fetch the details of a policy version

**Arguments****versionid** (Required)

The policy version ID

**General structure**

```
int //The policy version ID
```

**XML-RPC (PHP structure)**

```
[versionid] => int
```

**REST (POST parameters)**

```
versionid= int
```

**behalfid** (Default to "0")

The id of user on whose behalf the user is viewing the policy

**General structure**

```
int Default to "0" //The id of user on whose behalf the user is viewing the policy
```

**XML-RPC (PHP structure)**

```
[behalfid] => int
```

**REST (POST parameters)**

```
behalfid= int
```

**Response****General structure**

```
object {
 result object {
 policy Optional //Policy information
 object {
 name string Optional //The policy version name
 versionid int Optional //The policy version id
 content string Optional //The policy version content
 }
 }
 warnings Optional //List of warnings
 list of (
 //warning
 object {
 item string Optional //item
 itemid int Optional //item id
 warningcode string //the warning code can be used by the client app to implement specific behaviour
 message string //untranslated english message to explain the warning
 }
)
}
```

**XML-RPC (PHP structure)**

```

Array
(
 [
 [result] =>
 Array
 (
 [
 [policy] =>
 Array
 (
 [
 [name] => string
 [versionid] => int
 [content] => string
)
)
)
 [
 [warnings] =>
 Array
 (
 [
 [0] =>
 Array
 (
 [
 [item] => string
 [itemid] => int
 [warningcode] => string
 [message] => string
)
)
)
)
)
)

```

**REST**

```

<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="result">
 <SINGLE>
 <KEY name="policy">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="versionid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
 </KEY>
 <KEY name="warnings">
 <MULTIPLE>
 <SINGLE>
 <KEY name="item">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="itemid">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="warningcode">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="message">
 <VALUE>string</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 </SINGLE>
</RESPONSE>

```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

**Callable from AJAX**

Yes

**tool\_policy\_submit\_accept\_on\_behalf**

Accept policies on behalf of other users

**Arguments****jsonformdata** (Required)

The data from the create group form, encoded as a json array

**General structure**

```
string //The data from the create group form, encoded as a json array
```

**XML-RPC (PHP structure)**

```
[jsonformdata] => string
```

**REST (POST parameters)**

```
jsonformdata= string
```

**Response**

success

**General structure**

```
int //success
```

**XML-RPC (PHP structure)**

```
int
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <VALUE>int</VALUE>
</RESPONSE>
```

**Error message**

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_templatelibrary\_list\_templates**

List/search templates by component.

**Arguments****component** (Default to "")

The component to search

**General structure**

```
string Default to "" //The component to search
```

**XML-RPC (PHP structure)**

```
[component] => string
```

**REST (POST parameters)**

```
component= string
```

**search** (Default to "")

The search string

**General structure**

```
string Default to "" //The search string
```

**XML-RPC (PHP structure)**

```
[search] => string
```

**REST (POST parameters)**

```
search= string
```

**themename** (Default to "")

The current theme

**General structure**

```
string Default to "" //The current theme
```

**XML-RPC (PHP structure)**

```
[themename] => string
```

**REST (POST parameters)**

```
themename= string
```

**Response****General structure**

```
list of (
string //The template name (format is component/templatename)
)
```

**XML-RPC (PHP structure)**

```
Array
(
[0] => string
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <MULTIPLE>
 <VALUE>string</VALUE>
 </MULTIPLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

No

**Callable from AJAX**

Yes

**tool\_templatelibrary\_load\_canonical\_template**

Load a canonical template by name (not the theme overridden one).

**Arguments**

**component** (Required)

component containing the template

**General structure**

```
string //component containing the template
```

**XML-RPC (PHP structure)**

```
[component] => string
```

**REST (POST parameters)**

```
component= string
```

**template (Required)**

name of the template

**General structure**

```
string //name of the template
```

**XML-RPC (PHP structure)**

```
[template] => string
```

**REST (POST parameters)**

```
template= string
```

**Response**

template

**General structure**

```
string //template
```

**XML-RPC (PHP structure)**

```
string
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <VALUE>string</VALUE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

No

Callable from AJAX

Yes

`tool_usertours_complete_tour`

Mark the specified tour as completed for the current user

#### Arguments

`tourid` (Required)

Tour ID

**General structure**

```
int //Tour ID
```

**XML-RPC (PHP structure)**

```
[tourid] => int
```

**REST (POST parameters)**

```
tourid= int
```

`context` (Required)

Context ID

**General structure**

```
int //Context ID
```

**XML-RPC (PHP structure)**

```
[context] => int
```

**REST (POST parameters)**

```
context= int
```



`pageurl` (Required)

Page URL

**General structure**

```
string //Page URL
```

**XML-RPC (PHP structure)**

```
[pageurl] => string
```

**REST (POST parameters)**

```
pageurl= string
```

`stepid` (Required)

Step ID

**General structure**

```
int //Step ID
```

**XML-RPC (PHP structure)**

```
[stepid] => int
```

**REST (POST parameters)**

```
stepid= int
```

**stepindex** (Required)

Step Number

**General structure**

```
int //Step Number
```

**XML-RPC (PHP structure)**

```
[stepindex] => int
```

**REST (POST parameters)**

```
stepindex= int
```

**Response****General structure**

```
object {
}
```

**XML-RPC (PHP structure)**

```
Array
(
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

Restricted to logged-in users

Yes

Callable from AJAX

Yes

`tool_usertours_fetch_and_start_tour`

Fetch the specified tour

#### Arguments

`tourid` (Required)

Tour ID

**General structure**

```
int //Tour ID
```

**XML-RPC (PHP structure)**

```
[tourid] => int
```

**REST (POST parameters)**

```
tourid= int
```

`context` (Required)

Context ID

**General structure**

```
int //Context ID
```

**XML-RPC (PHP structure)**

```
[context] => int
```

**REST (POST parameters)**

```
context= int
```



`pageurl` (Required)

Page URL

**General structure**

```
string //Page URL
```

**XML-RPC (PHP structure)**

```
[pageurl] => string
```

**REST (POST parameters)**

```
pageurl= string
```

**Response**

```
General structure

object {
tourconfig Optional //Tour config
object {
name string //Tour Name
steps list of (
object {
title string //Step Title
content string //Step Content
element string //Step Target
placement string //Step Placement
delay int Optional //Delay before showing the step (ms)
backdrop int Optional //Whether a backdrop should be used
reflex int Optional //Whether to move to the next step when the target element is clicked
orphan int Optional //Whether to display the step even if it could not be found
stepid int Optional //The actual ID of the step
}
)endtourlabel string //Label of the end tour button
displaystepnumbers int //display step number
}
}
```

**XML-RPC (PHP structure)**

```
Array
(
[tourconfig] =>
Array
(
[name] => string
[steps] =>
Array
(
[0] =>
Array
(
[title] => string
[content] => string
[element] => string
[placement] => string
[delay] => int
[backdrop] => int
[reflex] => int
[orphan] => int
[stepid] => int
)
)
[endtourlabel] => string
[displaystepnumbers] => int
)
)
```



**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="tourconfig">
 <SINGLE>
 <KEY name="name">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="steps">
 <MULTIPLE>
 <SINGLE>
 <KEY name="title">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="content">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="element">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="placement">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="delay">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="backdrop">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="reflex">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="orphan">
 <VALUE>int</VALUE>
 </KEY>
 <KEY name="stepid">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </MULTIPLE>
 </KEY>
 <KEY name="endtourlabel">
 <VALUE>string</VALUE>
 </KEY>
 <KEY name="displaystepnumbers">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_usertours\_reset\_tour**

Remove the specified tour

**Arguments****tourid** (Required)

Tour ID

**General structure**

int //Tour ID

**XML-RPC (PHP structure)**

[tourid] =&gt; int

**REST (POST parameters)**

tourid= int

**context** (Required)

Context ID

**General structure**

int //Context ID

**XML-RPC (PHP structure)**

[context] =&gt; int

**REST (POST parameters)**

context= int

**pageurl** (Required)

Current page location

**General structure**

string //Current page Location

**XML-RPC (PHP structure)**

[pageurl] =&gt; string

**REST (POST parameters)**

pageurl= string

**Response****General structure**

```
object {
 startTour int Optional //Tour ID
}
```

**XML-RPC (PHP structure)**

```
Array
(
 [startTour] => int
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 <KEY name="startTour">
 <VALUE>int</VALUE>
 </KEY>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_usertours\_step\_shown**

Mark the specified step as completed for the current user

**Arguments****tourid** (Required)

Tour ID

**General structure**

```
int //Tour ID
```

**XML-RPC (PHP structure)**

```
[tourid] => int
```

**REST (POST parameters)**

```
tourid= int
```

**context** (Required)

Context ID

**General structure**

```
int //Context ID
```

**XML-RPC (PHP structure)**

```
[context] => int
```

**REST (POST parameters)**

```
context= int
```

pageurl (Required)

Page URL

**General structure**

```
string //Page URL
```

**XML-RPC (PHP structure)**

```
[pageurl] => string
```

**REST (POST parameters)**

```
pageurl= string
```

stepid (Required)

Step ID

**General structure**

```
int //Step ID
```

**XML-RPC (PHP structure)**

```
[stepid] => int
```

**REST (POST parameters)**

```
stepid= int
```



stepindex (Required)

Step Number

**General structure**

```
int //Step Number
```

**XML-RPC (PHP structure)**

```
[stepindex] => int
```

**REST (POST parameters)**

```
stepindex= int
```

**Response****General structure**

```
object {
}
```

**XML-RPC (PHP structure)**

```
Array
(
)
```

**REST**

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
 <SINGLE>
 </SINGLE>
</RESPONSE>
```

**Error message****REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```

**Restricted to logged-in users**

Yes

**Callable from AJAX**

Yes

**tool\_xmldb\_invoke\_move\_action**

moves element up/down

**Arguments****action** (Required)

Action

**General structure**

```
string //Action
```

**XML-RPC (PHP structure)**

```
[action] => string
```

**REST (POST parameters)**

```
action= string
```

dir (Required)

Plugin that is being edited

**General structure**

```
string //Plugin that is being edited
```

**XML-RPC (PHP structure)**

```
[dir] => string
```

**REST (POST parameters)**

```
dir= string
```

table (Required)

Table name

**General structure**

```
string //Table name
```

**XML-RPC (PHP structure)**

```
[table] => string
```

**REST (POST parameters)**

```
table= string
```

field (Default to "")

Field name

**General structure**

```
string Default to "" //Field name
```

**XML-RPC (PHP structure)**

```
[field] => string
```

**REST (POST parameters)**

```
field= string
```



key (Default to "")

Key name

**General structure**

```
string Default to "" //Key name
```

**XML-RPC (PHP structure)**

```
[key] => string
```

**REST (POST parameters)**

```
key= string
```

index (Default to "")

Index name

**General structure**

```
string Default to "" //Index name
```

**XML-RPC (PHP structure)**

```
[index] => string
```

**REST (POST parameters)**

```
index= string
```

**position** (Required)

How many positions to move by (negative - up, positive - down)

**General structure**

```
int //How many positions to move by (negative - up, positive - down)
```

**XML-RPC (PHP structure)**

```
[position] => int
```

**REST (POST parameters)**

```
position= int
```

## Response

### Error message

**REST**

```
<?xml version="1.0" encoding="UTF-8"?>
<EXCEPTION class="invalid_parameter_exception">
 <MESSAGE>Invalid parameter value detected</MESSAGE>
 <DEBUGINFO></DEBUGINFO>
</EXCEPTION>
```



### Restricted to logged-in users

Yes

### Callable from AJAX

Yes

**E-exam****Info**

Amhara Bank Website

Internal Menu

**Contact Us**

Addis Abeba

Phone : 690

E-mail : info@amharabank.com.et

**Follow Us**

Copyright © 2023

^