moodle

Admin Nutzer

moodle

Dashboard ▶ Site administration ▶ Plugins ▶ Web services ▶ API Documentation

Blocks editing on

NAVIGATION

Dashboard
- □ Site home
  - Site pages
  - Courses

ADMINISTRATION

- Site administration
  - □ Notifications
  - □ Registration
  - □ Advanced features
  - Users
  - Courses
  - Grades
  - Badges
  - Location
  - Language
  - Plugins
    - □ Install plugins
    - □ Plugins overview
    - Activity modules
    - Admin tools
    - Authentication
    - Blocks
    - Caching
    - Course formats
    - Enrolments
    - Filters
    - Licences
    - Local plugins
    - Logging
    - Message outputs
    - Question behaviours
    - Question types
    - Reports
    - Repositories
    - Text editors
    - Web services
      - □ Mobile
      - □ Overview
      - □ **API Documentation**
      - □ External services
      - □ Manage protocols
      - □ Manage tokens
  - Security
  - Appearance
  - Front page
  - Server

API Documentation

To create a client we advise you to read the Moodle web service client documentation

Print all

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 = MA
RKDOWN)
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 "1457727462" //timestart
timeduration int  Default to "0" //time duration
visible int  Default to "1" //visible
sequence int  Default to "1" //sequence
}
)
```

Reports

Development

□ Assignment upgrade helper

Search

ADMIN BOOKMARKS

Bookmark this page

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>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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 comeplete 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
No

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
)}
```

XML-RPC (PHP structure)

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

REST (POST parameters)

```
events[eventids][0]= int
events[courseids][0]= int
events[groupids][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 global 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
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
                        [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>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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 (alphanumeri
c 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 valu
e 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 val
ue 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 = PL
AIN or 4 = MARKDOWN)
visible int  Optional //cohort visible
}
)
```

```
XML-RPC (PHP structure)

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

```
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
```

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 = MARKD
OWN)
visible int   //cohort visible
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [name] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [visible] => 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="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>
        </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_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_delete_cohort_members

Deletes cohort members.

Arguments
members (Required)

```
General structure

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

```
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_get_cohorts

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 = MARKD
OWN)
visible int   //cohort visible
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [name] => string
                [idnumber] => string
                [description] => string
                [descriptionformat] => int
                [visible] => 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="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>
        </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_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_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 val
ue 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 = PL
AIN or 4 = MARKDOWN)
visible int  Optional //cohort visible
}
)
```

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
                )
        )
```

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
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<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_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
```

Response

General structure

```
object {
comments   //List of comments
list of ( 
  //comment
object {
id int   //Comment ID
content string   //The content text formated
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
} 
)warnings  Optional //list of warnings
list of ( 
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                        )
                )
        [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="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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   //comment ID
modname string   //activity module name
instance int   //instance ID
state int   //completion state value:
                                                                    0 means incomplete, 1 co
mplete,
                                                                    2 complete pass, 3 compl
ete fail
timecompleted int   //timestamp for completed activity
tracking int   //type of tracking:
                                                                    0 means none, 1 manual,
2 automatic
}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                        )
                )
        [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>
                </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_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>
                    </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

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
No

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 "weeks" //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
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 "0" //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 "0" //no group, separate, visible
groupmodeforce int  Default to "0" //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
} 
)} 
)
```

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
                [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
                                )
                        )
                )
        )
```

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][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
```

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) without user data

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 "1")

0

```
General structure

list of ( 
object {
name string   //The backup option name:
                                            "activities" (int) Include course activites (def
ault 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 i
s equal to no),
                                            "role_assignments" (int) Include role assignment
s  (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 comp
letion 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  (defa
ult 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_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) th
e category id,"name" (string) the category name,"parent" (int) the parent category id,"idnum
ber" (string) category idnumber - user must have 'moodle/category:manage' to search on idnum
ber,"visible" (int) whether the returned categories must be visible or hidden. If the key is
not passed,
                                              then the function return all categories that th
e user can see. - user must have 'moodle/category:manage' or 'moodle/category:viewhiddencate
gories' 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 = MARKD
OWN)
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)
                                              sectionid (int) Return only this section
                                              sectionnumber (int) Return only this section
with number (order)
                                              cmid (int) Return only this module informati
on (among the whole sections structure)
                                              modname (string) Return only modules with th
is 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 personaly
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)
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
description string  Optional //activity description
visible int  Optional //is the module visible
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 identation in the site
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
userid int   //User who added this content to moodle
author string   //Content owner
license string   //Content license
}
)}
)}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [name] => string
                [visible] => int
                [summary] => string
                [summaryformat] => int
                [modules] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [id] => int
                                [url] => string
                                [name] => string
                                [instance] => int
                                [description] => string
                                [visible] => int
                                [modicon] => string
                                [modname] => string
                                [modplural] => string
                                [availability] => string
                                [indent] => int
                                [contents] =>
                                    Array
                                        (
                                        [0] =>
                                            Array
                                                (
                                                [type] => string
                                                [filename] => string
                                                [filepath] => string
                                                [filesize] => int
                                                [fileurl] => string
                                                [content] => string
                                                [timecreated] => int
                                                [timemodified] => int
                                                [sortorder] => int
                                                [userid] => int
                                                [author] => string
                                                [license] => 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="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="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="visible">
                <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="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="userid">
                                        <VALUE>int</VALUE>
                                    </KEY>
                                    <KEY name="author">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                    <KEY name="license">
                                        <VALUE>string</VALUE>
                                    </KEY>
                                </SINGLE>
                            </MULTIPLE>
                        </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_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
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
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 secti
ons 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  Optional //timestamp when the course have been created
timemodified int  Optional //timestamp when the course have been modified
enablecompletion int  Optional //Enabled, control via completion and activity settings. D
isbaled,
                                    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
}
)}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [shortname] => string
                [categoryid] => int
                [categorysortorder] => int
                [fullname] => string
                [idnumber] => string
                [summary] => string
                [summaryformat] => int
                [format] => string
                [showgrades] => int
                [newsitems] => int
                [startdate] => int
                [numsections] => int
                [maxbytes] => int
                [showreports] => int
                [visible] => int
                [hiddensections] => int
                [groupmode] => int
                [groupmodeforce] => int
                [defaultgroupingid] => int
                [timecreated] => int
                [timemodified] => int
                [enablecompletion] => int
                [completionnotify] => int
                [lang] => string
                [forcetheme] => string
                [courseformatoptions] =>
                    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="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="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="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 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>
        </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_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 (defa
ult to 0 = No)
```

```
XML-RPC (PHP structure)

[deletecontent] => int
```

```
REST (POST parameters)

deletecontent= int
```

options (Default to "0")

0

```
General structure

list of (
object {
name string   //The backup option name:
                                           "activities" (int) Include course activites (def
ault 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_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 = PL
AIN 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
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
} 
)} 
)
```

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
                [newsitems] => int
                [startdate] => 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
                                )
                        )
                )
        )
```

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][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
```

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_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. * userfields ('string, string, ...') return only the values of these user fields. * limitfrom (integer) sql limit from. * limitnumber (integer) maximum number of returned users.

```
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 c
ourse has groups enabled and this param
                                                isn't defined, returns all the viewable user
s.
                                                This option requires 'moodle/site:accessallg
roups' 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 cour
se context.
                            * userfields ('string, string, ...') return only the values of t
hese user fields.
                            * limitfrom (integer) sql limit from.
                            * limitnumber (integer) maximum number of returned users.
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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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)
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
url string  Optional //URL 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 clas
s 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 = MARKD
OWN)
}
)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 course
s 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
                [icq] => string
                [skype] => string
                [yahoo] => string
                [aim] => string
                [msn] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [url] => 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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="city">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <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>
                        <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. Requir
es 'moodle/course:enrolreview' .
                            * userfields ('string, string, ...') return only the values of t
hese user fields.
                            * limitfrom (integer) sql limit from.
                            * limitnumber (integer) max number of users per course and capab
ility.
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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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)
description string  Optional //User profile description
descriptionformat int  Optional //User profile description format
city string  Optional //Home city of the user
url string  Optional //URL 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
                            [icq] => string
                            [skype] => string
                            [yahoo] => string
                            [aim] => string
                            [msn] => string
                            [department] => string
                            [institution] => string
                            [interests] => string
                            [firstaccess] => int
                            [lastaccess] => int
                            [description] => string
                            [descriptionformat] => int
                            [city] => string
                            [url] => 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="icq">
    <VALUE>string</VALUE>
</KEY>
<KEY name="skype">
    <VALUE>string</VALUE>
</KEY>
<KEY name="yahoo">
    <VALUE>string</VALUE>
</KEY>
<KEY name="aim">
    <VALUE>string</VALUE>
</KEY>
<KEY name="msn">
    <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="description">
    <VALUE>string</VALUE>
</KEY>
<KEY name="descriptionformat">
    <VALUE>int</VALUE>
</KEY>
<KEY name="city">
    <VALUE>string</VALUE>
</KEY>
<KEY name="url">
    <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>
    </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_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
```

Response

General structure

```
list of (
object {
id int   //id of course
shortname string   //short name of course
fullname string   //long name of course
enrolledusercount int   //Number of enrolled users in this course
idnumber string   //id number of course
visible int   //1 means visible, 0 means hidden 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
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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [shortname] => string
                [fullname] => string
                [enrolledusercount] => int
                [idnumber] => string
                [visible] => int
                [summary] => string
                [summaryformat] => int
                [format] => string
                [showgrades] => int
                [lang] => string
                [enablecompletion] => 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="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="showgrades">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="lang">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="enablecompletion">
                <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_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>
```

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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_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 thi
s 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
                                    - 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)

```
[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_grades_get_grades

Returns student course total grade and grades for activities. This function does not return category or manual items. This function is suitable for managers or teachers not students.

Arguments

courseid (Required)

id of course

```
General structure

int   //id of course
```

```
XML-RPC (PHP structure)

[courseid] => int
```

```
REST (POST parameters)

courseid= int
```

component (Default to "")

A component, for example mod_forum or mod_quiz

```
General structure

string  Default to "" //A component, for example mod_forum or mod_quiz
```

```
XML-RPC (PHP structure)

[component] => string
```

```
REST (POST parameters)

component= string
```

activityid (Default to "null")

The activity ID

```
General structure

int  Default to "null" //The activity ID
```

```
XML-RPC (PHP structure)

[activityid] => int
```

```
REST (POST parameters)

activityid= int
```

userids (Default to "Array ( ) ")

An array of user IDs, leave empty to just retrieve grade item information

```
General structure

 Default to "Array
(
)
" //An array of user IDs, leave empty to just retrieve grade item information
list of (
int   //user ID
)
```

```
XML-RPC (PHP structure)

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

REST (POST parameters)

```
userids[0]= int
```

Response

General structure

```
object {
items list of (
object {
activityid string   //The ID of the activity or "course" for the course grade item
itemnumber int   //Will be 0 unless the module has multiple grades
scaleid int   //The ID of the custom scale or 0
name string   //The module name
grademin double   //Minimum grade
grademax double   //Maximum grade
gradepass double   //The passing grade threshold
locked int   //0 means not locked, > 1 is a date to lock until
hidden int   //0 means not hidden, > 1 is a date to hide until
grades list of (
object {
userid int   //Student ID
grade double   //Student grade
locked int   //0 means not locked, > 1 is a date to lock until
hidden int   //0 means not hidden, 1 hidden, > 1 is a date to hide until
overridden int   //0 means not overridden, > 1 means overridden
feedback string   //Feedback from the grader
feedbackformat int   //The format of the feedback
usermodified int   //The ID of the last user to modify this student grade
datesubmitted int   //A timestamp indicating when the student submitted the activity
dategraded int   //A timestamp indicating when the assignment was grades
str_grade string   //A string representation of the grade
str_long_grade string   //A nicely formatted string representation of the grade
str_feedback string   //A formatted string representation of the feedback from the grader
}
)}
)outcomes  Optional //An array of outcomes associated with the grade items
list of (
object {
activityid string   //The ID of the activity or "course" for the course grade item
itemnumber int   //Will be 0 unless the module has multiple grades
scaleid int   //The ID of the custom scale or 0
name string   //The module name
locked int   //0 means not locked, > 1 is a date to lock until
hidden int   //0 means not hidden, > 1 is a date to hide until
grades list of (
object {
userid int   //Student ID
grade double   //Student grade
locked int   //0 means not locked, > 1 is a date to lock until
hidden int   //0 means not hidden, 1 hidden, > 1 is a date to hide until
feedback string   //Feedback from the grader
feedbackformat int   //The feedback format
usermodified int   //The ID of the last user to modify this student grade
str_grade string   //A string representation of the grade
str_feedback string   //A formatted string representation of the feedback from the grader
}
)}
)}
```

XML-RPC (PHP structure)

```
Array
    (
    [items] =>
        Array
            (
            [0] =>
                Array
                    (
                    [activityid] => string
                    [itemnumber] => int
                    [scaleid] => int
                    [name] => string
                    [grademin] => double
                    [grademax] => double
                    [gradepass] => double
                    [locked] => int
                    [hidden] => int
                    [grades] =>
                        Array
                            (
                            [0] =>
                                Array
                                    (
                                    [userid] => int
                                    [grade] => double
                                    [locked] => int
                                    [hidden] => int
                                    [overridden] => int
                                    [feedback] => string
                                    [feedbackformat] => int
                                    [usermodified] => int
                                    [datesubmitted] => int
                                    [dategraded] => int
                                    [str_grade] => string
                                    [str_long_grade] => string
                                    [str_feedback] => string
                                    )
                            )
                    )
            )
    [outcomes] =>
        Array
            (
            [0] =>
                Array
                    (
                    [activityid] => string
                    [itemnumber] => int
                    [scaleid] => int
                    [name] => string
                    [locked] => int
                    [hidden] => int
                    [grades] =>
                        Array
                            (
                            [0] =>
                                Array
                                    (
                                    [userid] => int
                                    [grade] => double
                                    [locked] => int
                                    [hidden] => int
                                    [feedback] => string
                                    [feedbackformat] => int
                                    [usermodified] => int
                                    [str_grade] => string
                                    [str_feedback] => string
```

```
                                        )
                                )
                        )
                )
        )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="items">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="activityid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemnumber">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="scaleid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="grademin">
                        <VALUE>double</VALUE>
                    </KEY>
                    <KEY name="grademax">
                        <VALUE>double</VALUE>
                    </KEY>
                    <KEY name="gradepass">
                        <VALUE>double</VALUE>
                    </KEY>
                    <KEY name="locked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="hidden">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="grades">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="grade">
                                    <VALUE>double</VALUE>
                                </KEY>
                                <KEY name="locked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="hidden">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="overridden">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="feedback">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="feedbackformat">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="usermodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="datesubmitted">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="dategraded">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="str_grade">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="str_long_grade">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="str_feedback">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="outcomes">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="activityid">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemnumber">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="scaleid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="name">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="locked">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="hidden">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="grades">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="userid">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="grade">
                                    <VALUE>double</VALUE>
                                </KEY>
                                <KEY name="locked">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="hidden">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="feedback">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="feedbackformat">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="usermodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="str_grade">
                                    <VALUE>string</VALUE>
                                </KEY>
```

```
                                <KEY name="str_feedback">
                                    <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

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_grade_get_definitions

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_grading_get_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] => stri
ng

                                                                [descriptionmarkersformat] =
> int

                                                                [maxscore] => double
                                                                )
                                                        )
                                                [guide_comments] =>
                                                    Array
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [id] => int
                                                                [sortorder] => int
                                                                [description] => string
                                                                [descriptionformat] => int
                                                                )
                                                        )
                                                )
                                        [rubric] =>
                                            Array
                                                (
                                                [rubric_criteria] =>
                                                    Array
```

```
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [id] => int
                                                                [sortorder] => int
                                                                [description] => string
                                                                [descriptionformat] => int
                                                                [levels] =>
                                                                    Array
                                                                        (
                                                                        [0] =>
                                                                            Array
                                                                                (
                                                                                [id] => int
                                                                                [score] => d
ouble
                                                                                [definition]
=> string

[definitionformat] => 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="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>
                                <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>
                                </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_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] => stri
ng
                                                                [descriptionmarkersformat] =
> int
                                                                [maxscore] => double
                                                                )
                                                        )
                                                [guide_comments] =>
                                                    Array
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [id] => int
                                                                [sortorder] => int
                                                                [description] => string
                                                                [descriptionformat] => int
                                                                )
                                                        )
                                                )
                                        [rubric] =>
                                            Array
                                                (
                                                [rubric_criteria] =>
                                                    Array
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [id] => int
                                                                [sortorder] => int
                                                                [description] => string
                                                                [descriptionformat] => int
                                                                [levels] =>
                                                                    Array
                                                                        (
                                                                        [0] =>
                                                                            Array
                                                                                (
                                                                                [id] => int
                                                                                [score] => d
ouble

                                                                                [definition]
=> string

[definitionformat] => 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="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>
                                <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>
                                </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 = MA
RKDOWN)
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 = MARKDOW
N)
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 = MARKDOW
N)
}
)}
}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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>
                    <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>
            </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

General structure

```
  //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
                                                        )
                                                )
                                        [guide_comments] =>
                                            Array
                                                (
                                                [0] =>
                                                    Array
                                                        (
                                                        [id] => int
                                                        [sortorder] => int
                                                        [description] => string
                                                        [descriptionformat] => int
                                                        )
                                                )
                                        )
                                [rubric] =>
                                    Array
                                        (
                                        [rubric_criteria] =>
                                            Array
                                                (
                                                [0] =>
                                                    Array
                                                        (
                                                        [id] => int
                                                        [sortorder] => int
                                                        [description] => string
                                                        [descriptionformat] => int
                                                        [levels] =>
                                                            Array
                                                                (
                                                                [0] =>
                                                                    Array
                                                                        (
                                                                        [id] => int
                                                                        [score] => double
                                                                        [definition] => stri
ng
                                                                        [definitionformat] =
> int
                                                                        )
                                                                )
                                                        )
                                                )
                                        )
                                )
                        )
                )
        )
```

```
REST (POST parameters)

areas[0][cmid]= 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)
}
)
```

XML-RPC (PHP structure)

```
[groupings] =>
    Array
        (
        [0] =>
            Array
                (
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                )
        )
```

REST (POST parameters)

```
groupings[0][courseid]= int
groupings[0][name]= string
groupings[0][description]= string
groupings[0][descriptionformat]= int
```

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)
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => 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>
        </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
}
)
```

```
XML-RPC (PHP structure)

[groups] =>
    Array
        (
        [0] =>
            Array
                (
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => string
                )
        )
```

```
REST (POST parameters)

groups[0][courseid]= int
groups[0][name]= string
groups[0][description]= string
groups[0][descriptionformat]= int
groups[0][enrolmentkey]= string
```

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
}
)
```

XML-RPC (PHP structure)

```

    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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_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_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_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 = MARKD
OWN)
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => 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>
        </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   //multilang compatible name, course unique
description string   //group description text
descriptionformat int   //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARKD
OWN)
enrolmentkey string   //group enrol secret phrase
}
)
```

```
XML-RPC (PHP structure)


    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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_user_groups

Returns all groups in specified course for the specified user.

Arguments
courseid (Required)
id of course

```
General structure

int   //id of course
```

XML-RPC (PHP structure)

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

REST (POST parameters)

```
courseid= int
```

userid (Required)

id of user

General structure

```
int   //id of 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   //multilang compatible name, course unique
description string   //group description text
descriptionformat int   //description 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
        (
        [groups] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [id] => int
                        [name] => string
                        [description] => string
                        [descriptionformat] => 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>
                </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_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
```

returngroups (Default to "0")

return associated groups

General structure

```
int  Default to "0" //return associated groups
```

XML-RPC (PHP structure)

```
[returngroups] => int
```

REST (POST parameters)

```
returngroups= 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)
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
}
)}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [groups] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [id] => int
                                [courseid] => int
                                [name] => string
                                [description] => string
                                [descriptionformat] => int
                                [enrolmentkey] => 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="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>
                    </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   //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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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_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_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)
}
)
```

```
XML-RPC (PHP structure)

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

```
REST (POST parameters)

groupings[0][id]= int
groupings[0][name]= string
groupings[0][description]= string
groupings[0][descriptionformat]= 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_message_block_contacts

Block contacts

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
```

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

core_message_create_contacts

Add contacts to 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
```

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

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
```

Response

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<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_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_contacts

Retrieve the contact list

Arguments

Response

```
General structure

object {
online   //List of online 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
unread int   //Unread message count
}
)offline   //List of offline 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
unread int   //Unread message count
}
)strangers   //List of users that are not in the user's contact list but have sent a message
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
unread int   //Unread message count
}
)}
```

XML-RPC (PHP structure)

```
    Array
        (
        [online] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [id] => int
                        [fullname] => string
                        [profileimageurl] => string
                        [profileimageurlsmall] => string
                        [unread] => int
                        )
                )
        [offline] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [id] => int
                        [fullname] => string
                        [profileimageurl] => string
                        [profileimageurlsmall] => string
                        [unread] => int
                        )
                )
        [strangers] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [id] => int
                        [fullname] => string
                        [profileimageurl] => string
                        [profileimageurlsmall] => string
                        [unread] => int
                        )
                )
        )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <SINGLE>
        <KEY name="online">
            <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>
                    <KEY name="unread">
```

```
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="offline">
            <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>
                    <KEY name="unread">
                        <VALUE>int</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
        <KEY name="strangers">
            <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>
                    <KEY name="unread">
                        <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
No

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")

true for getting read messages, false for unread

```
General structure

int  Default to "1" //true for getting read messages, false for unread
```

```
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 = MARKD
OWN)
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
usertofullname string   //User to full name
userfromfullname string   //User from full 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
        (
        [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
                        [usertofullname] => string
                        [userfromfullname] => 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>
                </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_mark_message_read

Mark a single message as read, trigger message_viewed event.

Arguments

messageid (Required)

id of the message (in the message table)

```
General structure

int   //id of the message (in the message table)
```

```
XML-RPC (PHP structure)

[messageid] => int
```

```
REST (POST parameters)

messageid= int
```

timeread (Required)

timestamp for when the message should be marked read

```
General structure

int   //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 message_read 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
No

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 "1" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARK
DOWN)
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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [msgid] => int
                [clientmsgid] => string
                [errormessage] => string
                )
        )
```

```
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>
        </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_unblock_contacts

Unblock contacts

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
```

Response

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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, t
he 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
No

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)
}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                        )
                )
        [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>
                    <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="warnings">
            <MULTIPLE>
                <SINGLE>
                    <KEY name="item">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="itemid">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="warningcode">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="message">
                        <VALUE>string</VALUE>
                    </KEY>
                </SINGLE>
            </MULTIPLE>
        </KEY>
    </SINGLE>
</RESPONSE>
```

Error message

REST

```
<?xml version="1.0" encoding="UTF-8"?>
<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 e
rrorcode 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_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
```

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_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_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
                                    (block, course, coursecat, system, user, module)
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_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
```

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_create_users

Create users.

Arguments

users (Required)

General structure

```
list of (
object {
username string   //Username policy is defined in Moodle security config.
password string   //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
auth string  Default to "manual" //Auth plugins include manual, ldap, imap, etc
idnumber string  Default to "" //An arbitrary ID code number perhaps from the institution
lang string  Default to "de" //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
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, no HTML
city string  Optional //Home city of the user
country string  Optional //Home country code of the user, such as AU or CZ
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
preferences  Optional //User preferences
list of (
object {
type string   //The name of the preference
value string   //The value of the preference
}
)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
}
)}
)
```

XML-RPC (PHP structure)

```
[users] =>
    Array 
        (
        [0] =>
            Array 
                (
                [username] => string
                [password] => string
                [firstname] => string
                [lastname] => string
                [email] => string
                [auth] => string
                [idnumber] => string
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [city] => string
                [country] => string
                [firstnamephonetic] => string
                [lastnamephonetic] => string
                [middlename] => string
                [alternatename] => string
                [preferences] =>
                    Array 
                        (
                        [0] =>
                            Array 
                                (
                                [type] => string
                                [value] => string
                                )
                        )
                [customfields] =>
                    Array 
                        (
                        [0] =>
                            Array 
                                (
                                [type] => string
                                [value] => string
                                )
                        )
                )
        )
```

REST (POST parameters)

```
users[0][username]= string
users[0][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][auth]= string
users[0][idnumber]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][timezone]= string
users[0][mailformat]= int
users[0][description]= string
users[0][city]= string
users[0][country]= string
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
users[0][preferences][0][type]= string
users[0][preferences][0][value]= string
users[0][customfields][0][type]= string
users[0][customfields][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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 
4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
name string   //The name of the custom field
shortname string   //The shortname of the custom field - to be able to build the field clas
s in the code
} 
)preferences  Optional //Users preferences
list of ( 
object {
name string   //The name of the preferences
value string   //The value of the custom field
} 
)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 = MARKD
OWN)
}
)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 course
s 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
                [icq] => string
                [skype] => string
                [yahoo] => string
                [aim] => string
                [msn] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [url] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
```

```
                        (
                        [0] =>
                            Array
                                (
                                [type] => string
                                [value] => 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="icq">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="skype">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="yahoo">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="aim">
                    <VALUE>string</VALUE>
                </KEY>
                <KEY name="msn">
                    <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="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="url">
                    <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="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_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 occurences 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.

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 => 'manu
al', ...) -
                        key occurences are forbidden.
                        The search is executed with AND operator on the criterias. Invalid c
riterias (keys) are ignored,
                        the search is still executed on the valid criterias.
                        You can search without criteria, but the function is not designed fo
r it.
                        It could very slow or timeout. The function is designed to search so
me 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 % fo
r 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 searchi
ng 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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or
4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
name string   //The name of the custom field
shortname string   //The shortname of the custom field - to be able to build the field clas
s in the code
}
)preferences  Optional //Users preferences
list of (
object {
name string   //The name of the preferences
value string   //The value of the custom field
}
)}
)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
                    [icq] => string
                    [skype] => string
                    [yahoo] => string
                    [aim] => string
                    [msn] => string
                    [department] => string
                    [institution] => string
                    [idnumber] => string
                    [interests] => string
                    [firstaccess] => int
                    [lastaccess] => int
                    [auth] => string
                    [confirmed] => int
                    [lang] => string
                    [calendartype] => string
                    [theme] => string
                    [timezone] => string
                    [mailformat] => int
                    [description] => string
                    [descriptionformat] => int
                    [city] => string
                    [url] => string
                    [country] => string
                    [profileimageurlsmall] => string
                    [profileimageurl] => string
                    [customfields] =>
                        Array
                            (
                            [0] =>
                                Array
                                    (
                                    [type] => string
                                    [value] => 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="icq">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="skype">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="yahoo">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="aim">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="msn">
                        <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="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="url">
    <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="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
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()

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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or
4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
name string   //The name of the custom field
shortname string   //The shortname of the custom field - to be able to build the field clas
s in the code
}
)preferences  Optional //Users preferences
list of (
object {
name string   //The name of the preferences
value string   //The value of the custom field
}
)}
)
```

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
                [icq] => string
                [skype] => string
                [yahoo] => string
                [aim] => string
                [msn] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [url] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [type] => string
                                [value] => 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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="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="url">
                <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="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
No

core_user_get_users_by_id

DEPRECATED: this deprecated function will be removed in a future version. This function has been replaced by core_user_get_users_by_field()

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

```
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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
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 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
            (
            [id] => int
            [username] => string
            [firstname] => string
            [lastname] => string
            [fullname] => string
            [email] => string
            [address] => string
            [phone1] => string
            [phone2] => string
            [icq] => string
            [skype] => string
            [yahoo] => string
            [aim] => string
            [msn] => string
            [department] => string
            [institution] => string
            [idnumber] => string
            [interests] => string
            [firstaccess] => int
            [lastaccess] => int
            [auth] => string
            [confirmed] => int
            [lang] => string
            [calendartype] => string
            [theme] => string
            [timezone] => string
            [mailformat] => int
            [description] => string
            [descriptionformat] => int
            [city] => string
            [url] => string
            [country] => string
            [profileimageurlsmall] => string
            [profileimageurl] => string
            [customfields] =>
                Array
                    (
                    [0] =>
                        Array
                            (
                            [type] => string
                            [value] => string
                            [name] => string
                            [shortname] => string
                            )
                    )
            [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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="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="url">
                <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="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_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_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.
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
auth string  Optional //Auth plugins include manual, ldap, imap, etc
idnumber string  Optional //An arbitrary ID code number perhaps from the institution
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, no HTML
city string  Optional //Home city of the user
country string  Optional //Home country code of the user, such as AU or CZ
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
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
                [password] => string
                [firstname] => string
                [lastname] => string
                [email] => string
                [auth] => string
                [idnumber] => string
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [city] => string
                [country] => string
                [firstnamephonetic] => string
                [lastnamephonetic] => string
                [middlename] => string
                [alternatename] => string
                [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][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][auth]= string
users[0][idnumber]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][timezone]= string
users[0][mailformat]= int
users[0][description]= string
users[0][city]= string
users[0][country]= string
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
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

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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   //user 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 forcelogin is s
et to NO and guestaccess is set to YES.
                   In order to retrieve user profile pictures independently of the Moodle c
onfig, 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 us
er can ignore the upload file size
}
```

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
        )
```

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

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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 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
```

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
celltype string   //cell type
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
}
contributiontocoursetotal  Optional //contributiontocoursetotal column
object {
class string   //class
content string   //cell content
headers string   //headers
}
}
)}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                                                [celltype] => 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
                                                )
                                        [feedback] =>
                                            Array
                                                (
                                                [class] => string
                                                [content] => string
                                                [headers] => string
                                                )
                                        [contributiontocoursetotal] =>
                                            Array
                                                (
                                                [class] => string
                                                [content] => string
                                                [headers] => 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="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="tabledata">
    <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="celltype">
                        <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>
                    <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="contributiontocoursetotal">
                                    <SINGLE>
                                        <KEY name="class">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="content">
                                            <VALUE>string</VALUE>
                                        </KEY>
                                        <KEY name="headers">
                                            <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
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

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_is_system_configured

Check whether the airnotifier settings have been configured

Arguments

Response

0 if the system is not configured, 1 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

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
)
```

```
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
```

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
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
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
configs   //configuration settings
list of (
  //assignment configuration object
object {
id int   //assign_plugin_config id
assignment int   //assignment id
plugin string   //plugin
subtype string   //subtype
name string   //name
value string   //value
}
)intro string  Optional //assignment intro, not allways returned because it deppends on the activity configuration
introformat int  Optional //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARKDOWN)
introattachments  Optional //intro attachments files
list of (
object {
filename string   //file name
mimetype string   //mime type
fileurl string   //file download url
}
)}
)}
)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
                                        [teamsubmission] => int
                                        [requireallteammemberssubmit] => int
                                        [teamsubmissiongroupingid] => int
                                        [blindmarking] => int
                                        [revealidentities] => int
                                        [attemptreopenmethod] => string
                                        [maxattempts] => int
                                        [markingworkflow] => int
                                        [markingallocation] => int
                                        [requiresubmissionstatement] => int
                                        [configs] =>
                                            Array
                                                (
                                                [0] =>
                                                    Array
                                                        (
                                                        [id] => int
                                                        [assignment] => int
                                                        [plugin] => string
                                                        [subtype] => string
                                                        [name] => string
                                                        [value] => string
                                                        )
                                                )
                                        [intro] => string
                                        [introformat] => int
                                        [introattachments] =>
                                            Array
                                                (
                                                [0] =>
                                                    Array
                                                        (
                                                        [filename] => string
                                                        [mimetype] => string
                                                        [fileurl] => 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="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="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>
<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="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="introattachments">
                                    <MULTIPLE>
                                        <SINGLE>
                                            <KEY name="filename">
                                                <VALUE>string</VALUE>
                                            </KEY>
                                            <KEY name="mimetype">
                                                <VALUE>string</VALUE>
                                            </KEY>
                                            <KEY name="fileurl">
                                                <VALUE>string</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

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 (
object {
id int   //grade id
userid int   //student id
attemptnumber int   //attempt number
timecreated int   //grade creation time
timemodified int   //grade last modified time
grader int   //grader
grade string   //grade
}
)}
)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
                                        [userid] => int
                                        [attemptnumber] => int
                                        [timecreated] => int
                                        [timemodified] => int
                                        [grader] => int
                                        [grade] => 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="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>
                            </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_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 (
object {
id int   //submission id
userid int   //student id
attemptnumber int   //attempt number
timecreated int   //submission creation time
timemodified int   //submission last modified time
status string   //submission status
groupid int   //group id
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 (
object {
filepath string   //file path
fileurl string  Optional //file download url
}
)}
)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)
}
)}
)}
)}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can be used by the client app to implement specific
behaviour
message string   //untranslated english message to explain the warning
}
)}
```

XML-RPC (PHP structure)

```
    Array
        (
        [assignments] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [assignmentid] => int
                        [submissions] =>
                            Array 
                                (
                                [0] =>
                                    Array 
                                        (
                                        [id] => int
                                        [userid] => int
                                        [attemptnumber] => int
                                        [timecreated] => int
                                        [timemodified] => int
                                        [status] => string
                                        [groupid] => int
                                        [plugins] =>
                                            Array 
                                                (
                                                [0] =>
                                                    Array 
                                                        (
                                                        [type] => string
                                                        [name] => string
                                                        [fileareas] =>
                                                            Array 
                                                                (
                                                                [0] =>
                                                                    Array 
                                                                        (
                                                                        [area] => string
                                                                        [files] =>
                                                                            Array 
                                                                                (
                                                                                [0] =>
                                                                                    Array 
                                                                                        (

[filepath] => string

[fileurl] => string
                                                                                        )
                                                                                )
                                                                        )
                                                                )
                                                        [editorfields] =>
                                                            Array 
                                                                (
                                                                [0] =>
                                                                    Array 
                                                                        (
                                                                        [name] => string
                                                                        [description] => str
ing
                                                                        [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="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="timemodified">
                                    <VALUE>int</VALUE>
                                </KEY>
                                <KEY name="status">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="groupid">
                                    <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="filepath">

<VALUE>string</VALUE>
                                                                    </KEY>
                                                                    <KEY name="fileurl">

<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>
                </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_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 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 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_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 assignmen
ts).
```

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 object {
text string   //The text for this feedback.
format int   //The format for this feedback
}
files_filemanager int   //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 = MARKDOW
N)
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 = MARKDOW
N)
}
)}
)}
}
```

XML-RPC (PHP structure)

```
[advancedgradingdata] =>
    Array
        (
        [guide] =>
            Array
                (
                [criteria] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [criterionid] => int
                                [fillings] =>
                                    Array
                                        (
                                        [0] =>
                                            Array
                                                (
                                                [criterionid] => int
                                                [levelid] => int
                                                [remark] => string
                                                [remarkformat] => int
                                                [score] => double
                                                )
                                        )
                                )
                        )
                )
        [rubric] =>
            Array
                (
                [criteria] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [criterionid] => int
                                [fillings] =>
                                    Array
                                        (
                                        [0] =>
                                            Array
                                                (
                                                [criterionid] => int
                                                [levelid] => int
                                                [remark] => string
                                                [remarkformat] => int
                                                )
                                        )
                                )
                        )
                )
        )
```

```
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 assignmen
ts).
```

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 object {
text string   //The text for this feedback.
format int   //The format for this feedback
}
files_filemanager int   //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 = MARKDOW
N)
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 = MARKDOW
N)
}
)}
)}
}
}
)
```

XML-RPC (PHP structure)

```
[grades] =>
    Array
        (
        [0] =>
            Array
                (
                [userid] => int
                [grade] => double
                [attemptnumber] => int
                [addattempt] => int
                [workflowstate] => string
                [plugindata] =>
                    Array
                        (
                        [assignfeedbackcomments_editor] =>
                            Array
                                (
                                [text] => string
                                [format] => int
                                )
                        [files_filemanager] => int
                        )
                [advancedgradingdata] =>
                    Array
                        (
                        [guide] =>
                            Array
                                (
                                [criteria] =>
                                    Array
                                        (
                                        [0] =>
                                            Array
                                                (
                                                [criterionid] => int
                                                [fillings] =>
                                                    Array
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [criterionid] => int
                                                                [levelid] => int
                                                                [remark] => string
                                                                [remarkformat] => int
                                                                [score] => double
                                                                )
                                                        )
                                                )
                                        )
                                )
                        [rubric] =>
                            Array
                                (
                                [criteria] =>
                                    Array
                                        (
                                        [0] =>
                                            Array
                                                (
                                                [criterionid] => int
                                                [fillings] =>
                                                    Array
                                                        (
                                                        [0] =>
                                                            Array
                                                                (
                                                                [criterionid] => int
                                                                [levelid] => int
                                                                [remark] => string
                                                                [remarkformat] => int
                                                                )
                                                        )
                                                )
                                        )
                                )
                        )
                )
        )
```

```
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 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   //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_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_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_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 ( 
  //Database
object {
id int   //Database id
coursemodule int   //Course module id
course int   //Course id
name string   //Database name
intro string   //The Database intro
introformat int   //intro format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARKDOWN)
comments int  Optional //comments enabled
timeavailablefrom int  Optional //timeavailablefrom field
timeavailableto int  Optional //timeavailableto field
timeviewfrom int  Optional //timeviewfrom field
timeviewto int  Optional //timeviewto field
requiredentries int  Optional //requiredentries field
requiredentriestoview int  Optional //requiredentriestoview field
maxentries int  Optional //maxentries field
rssarticles int  Optional //rssarticles field
singletemplate string  Optional //singletemplate field
listtemplate string  Optional //listtemplate field
listtemplateheader string  Optional //listtemplateheader field
listtemplatefooter string  Optional //listtemplatefooter field
addtemplate string  Optional //addtemplate field
rsstemplate string  Optional //rsstemplate field
rsstitletemplate string  Optional //rsstitletemplate field
csstemplate string  Optional //csstemplate field
jstemplate string  Optional //jstemplate field
asearchtemplate string  Optional //asearchtemplate field
approval int  Optional //approval field
scale int  Optional //scale field
assessed int  Optional //assessed field
assesstimestart int  Optional //assesstimestart field
assesstimefinish int  Optional //assesstimefinish field
defaultsort int  Optional //defaultsort field
defaultsortdir int  Optional //defaultsortdir field
editany int  Optional //editany field
notification int  Optional //notification 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
        (
        [databases] =>
            Array
                (
                [0] =>
                    Array
                        (
                        [id] => int
                        [coursemodule] => int
                        [course] => int
                        [name] => string
                        [intro] => string
                        [introformat] => int
                        [comments] => int
                        [timeavailablefrom] => int
                        [timeavailableto] => int
                        [timeviewfrom] => int
                        [timeviewto] => int
                        [requiredentries] => int
                        [requiredentriestoview] => int
                        [maxentries] => int
                        [rssarticles] => int
                        [singletemplate] => string
                        [listtemplate] => string
                        [listtemplateheader] => string
                        [listtemplatefooter] => string
                        [addtemplate] => string
                        [rsstemplate] => string
                        [rsstitletemplate] => string
                        [csstemplate] => string
                        [jstemplate] => string
                        [asearchtemplate] => string
                        [approval] => int
                        [scale] => int
                        [assessed] => int
                        [assesstimestart] => int
                        [assesstimefinish] => int
                        [defaultsort] => int
                        [defaultsortdir] => int
                        [editany] => int
                        [notification] => 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="databases">
```

```
            <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="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="requiredentriestoview">
                        <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>
                    <KEY name="approval">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="scale">
                        <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="defaultsort">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="defaultsortdir">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="editany">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="notification">
                        <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_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, 2 = PLAIN or 4 = MARKDOWN)
assessed int   //Aggregate type
assesstimestart int   //Assess start time
assesstimefinish int   //Assess finish time
scale int   //Scale
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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [course] => int
                [type] => string
                [name] => string
                [intro] => string
                [introformat] => int
                [assessed] => int
                [assesstimestart] => int
                [assesstimefinish] => int
                [scale] => 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
                )
        )
```

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="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="maxbytes">
                <VALUE>int</VALUE>
            </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>
        </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_get_forum_discussions

DEPRECATED (use mod_forum_get_forum_discussions_paginated instead): Returns a list of forum discussions contained within a given set of forums.

Arguments

forumids (Required)

Array of Forum IDs

```
General structure

  //Array of Forum IDs
list of (
int   //forum ID
)
```

```
XML-RPC (PHP structure)

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

```
REST (POST parameters)

forumids[0]= 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 (
  //discussion
object {
id int   //Forum id
course int   //Course id
forum int   //The forum id
name string   //Discussion name
userid int   //User id
groupid int   //Group id
assessed int   //Is this assessed?
timemodified int   //Time modified
usermodified int   //The id of the user who last modified
timestart int   //Time discussion can start
timeend int   //Time discussion ends
firstpost int   //The first post in the discussion
firstuserfullname string   //The discussion creators fullname
firstuserimagealt string   //The discussion creators image alt
firstuserpicture int   //The discussion creators profile picture
firstuseremail string   //The discussion creators email
subject string   //The discussion subject
numreplies string   //The number of replies in the discussion
numunread string   //The number of unread posts, blank if this value is
                       not available due to forum settings.
lastpost int   //The id of the last post in the discussion
lastuserid int   //The id of the user who made the last post
lastuserfullname string   //The last person to posts fullname
lastuserimagealt string   //The last person to posts image alt
lastuserpicture int   //The last person to posts profile picture
lastuseremail string   //The last person to posts email
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [course] => int
                [forum] => int
                [name] => string
                [userid] => int
                [groupid] => int
                [assessed] => int
                [timemodified] => int
                [usermodified] => int
                [timestart] => int
                [timeend] => int
                [firstpost] => int
                [firstuserfullname] => string
                [firstuserimagealt] => string
                [firstuserpicture] => int
                [firstuseremail] => string
                [subject] => string
                [numreplies] => string
                [numunread] => string
                [lastpost] => int
                [lastuserid] => int
                [lastuserfullname] => string
                [lastuserimagealt] => string
                [lastuserpicture] => int
                [lastuseremail] => string
                )
        )
```

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="forum">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="name">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="userid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="groupid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="assessed">
                <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="firstpost">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="firstuserfullname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="firstuserimagealt">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="firstuserpicture">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="firstuseremail">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="subject">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="numreplies">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="numunread">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="lastpost">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="lastuserid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="lastuserfullname">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="lastuserimagealt">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="lastuserpicture">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="lastuseremail">
                <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_forum_get_forum_discussions_paginated

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, timestart or timeend

```
General structure

string  Default to "timemodified" //sort by this element: id, timemodified, timestart 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
timestart 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?
attachment string   //Has attachments?
attachments  Optional //attachments
list of (
object {
filename string   //file name
mimetype string   //mime type
fileurl string   //file download url
}
)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 string   //The number of replies in the discussion
numunread int   //The number of unread discussions.
}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                        [attachment] => string
                        [attachments] =>
                            Array
                                (
                                [0] =>
                                    Array
                                        (
                                        [filename] => string
                                        [mimetype] => string
                                        [fileurl] => string
                                        )
                                )
                        [totalscore] => int
                        [mailnow] => int
                        [userfullname] => string
                        [usermodifiedfullname] => string
                        [userpictureurl] => string
                        [usermodifiedpictureurl] => string
                        [numreplies] => string
                        [numunread] => 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="attachment">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="attachments">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="fileurl">
```

```
                                    <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>string</VALUE>
                    </KEY>
                    <KEY name="numunread">
                        <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_discussion_posts

Returns a list of forum posts for a discussion.

Arguments

discussionid (Required)

discussion ID

```
General structure

int   //discussion ID
```

```
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
```

Response

```
General structure

object {
posts list of (
  //post
object {
id int   //Post id
discussion int   //Discussion id
parent int   //Parent id
userid int   //User 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?
attachment string   //Has attachments?
attachments  Optional //attachments
list of (
object {
filename string   //file name
mimetype string   //mime type
fileurl string   //file download url
}
)totalscore int   //The post message total score
mailnow int   //Mail now?
children list of (
int   //children post id
)canreply int   //The user can reply to posts?
postread int   //The post was read
userfullname string   //Post author full name
userpictureurl string  Optional //Post author picture.
}
)warnings  Optional //list of warnings
list of (
  //warning
object {
item string  Optional //item
itemid int  Optional //item id
warningcode string   //the warning code can 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
                        [discussion] => int
                        [parent] => int
                        [userid] => int
                        [created] => int
                        [modified] => int
                        [mailed] => int
                        [subject] => string
                        [message] => string
                        [messageformat] => int
                        [messagetrust] => int
                        [attachment] => string
                        [attachments] =>
                            Array
                                (
                                [0] =>
                                    Array
                                        (
                                        [filename] => string
                                        [mimetype] => string
                                        [fileurl] => string
                                        )
                                )
                        [totalscore] => int
                        [mailnow] => int
                        [children] =>
                            Array
                                (
                                [0] => int
                                )
                        [canreply] => int
                        [postread] => int
                        [userfullname] => string
                        [userpictureurl] => 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="posts">
```

```
            <MULTIPLE>
                <SINGLE>
                    <KEY name="id">
                        <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="attachment">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="attachments">
                        <MULTIPLE>
                            <SINGLE>
                                <KEY name="filename">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="mimetype">
                                    <VALUE>string</VALUE>
                                </KEY>
                                <KEY name="fileurl">
                                    <VALUE>string</VALUE>
                                </KEY>
                            </SINGLE>
                        </MULTIPLE>
                    </KEY>
                    <KEY name="totalscore">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="mailnow">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="children">
                        <MULTIPLE>
                            <VALUE>int</VALUE>
                        </MULTIPLE>
                    </KEY>
                    <KEY name="canreply">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="postread">
                        <VALUE>int</VALUE>
                    </KEY>
                    <KEY name="userfullname">
                        <VALUE>string</VALUE>
                    </KEY>
                    <KEY name="userpictureurl">
                        <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_forum_view_forum

Simulate the view.php web interface page: trigger events, completion, etc...

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

Simulate the forum/discuss.php web interface page: trigger events, completion, etc...

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

moodle_course_create_courses

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_course_create_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 "weeks" //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
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 "0" //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 secti
ons in the course are displayed to students
groupmode int  Default to "0" //no group, separate, visible
groupmodeforce int  Default to "0" //1: yes, 0: no
defaultgroupingid int  Default to "0" //default grouping id
enablecompletion int  Optional //Enabled, control via completion and activity settings. D
isabled,
                                    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
}
)}
)
```

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
                [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
                                )
                        )
                )
        )
```

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][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
```

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

moodle_course_get_courses

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_course_get_courses()

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
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
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  Optional //timestamp when the course have been created
timemodified int  Optional //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
}
)}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [shortname] => string
                [categoryid] => int
                [categorysortorder] => int
                [fullname] => string
                [idnumber] => string
                [summary] => string
                [summaryformat] => int
                [format] => string
                [showgrades] => int
                [newsitems] => int
                [startdate] => int
                [numsections] => int
                [maxbytes] => int
                [showreports] => int
                [visible] => int
                [hiddensections] => int
                [groupmode] => int
                [groupmodeforce] => int
                [defaultgroupingid] => int
                [timecreated] => int
                [timemodified] => int
                [enablecompletion] => int
                [completionnotify] => int
                [lang] => string
                [forcetheme] => string
                [courseformatoptions] =>
                    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="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="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="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 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>
        </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

moodle_enrol_get_enrolled_users

DEPRECATED: this deprecated function will be removed in a future version. Please use core_enrol_get_enrolled_users() (previously known as moodle_user_get_users_by_courseid).

Arguments

courseid (Required)

Course id

```
General structure

int   //Course id
```

```
XML-RPC (PHP structure)

[courseid] => int
```

```
REST (POST parameters)

courseid= int
```

withcapability (Default to "null")

User should have this capability

General structure

```
string  Default to "null" //User should have this capability
```

XML-RPC (PHP structure)

```
[withcapability] => string
```

REST (POST parameters)

```
withcapability= string
```

groupid (Default to "null")

Group id, null means all groups

General structure

```
int  Default to "null" //Group id, null means all groups
```

XML-RPC (PHP structure)

```
[groupid] => int
```

REST (POST parameters)

```
groupid= int
```

onlyactive (Default to "0")

True means only active, false means all participants

General structure

```
int  Default to "0" //True means only active, false means all participants
```

XML-RPC (PHP structure)

```
[onlyactive] => int
```

REST (POST parameters)

```
onlyactive= int
```

Response

General structure

```
list of (
object {
courseid int   //id of course
userid int   //id of user
firstname string  Optional //first name of user
lastname string  Optional //last name of user
fullname string   //fullname of user
username string  Optional //username of user
profileimgurl string   //url of the profile image
profileimgurlsmall string   //url of the profile image (small version)
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [courseid] => int
                [userid] => int
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [username] => string
                [profileimgurl] => string
                [profileimgurlsmall] => string
                )
        )
```

REST

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESPONSE>
    <MULTIPLE>
        <SINGLE>
            <KEY name="courseid">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="userid">
                <VALUE>int</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="username">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimgurl">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="profileimgurlsmall">
                <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

moodle_enrol_get_users_courses

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_enrol_get_users_courses()

Arguments

userid (Required)

user id

General structure

```
int   //user id
```

XML-RPC (PHP structure)

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

REST (POST parameters)

```
userid= int
```

Response

General structure

```
list of (
object {
id int   //id of course
shortname string   //short name of course
fullname string   //long name of course
enrolledusercount int   //Number of enrolled users in this course
idnumber string   //id number of course
visible int   //1 means visible, 0 means hidden 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
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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [shortname] => string
                [fullname] => string
                [enrolledusercount] => int
                [idnumber] => string
                [visible] => int
                [summary] => string
                [summaryformat] => int
                [format] => string
                [showgrades] => int
                [lang] => string
                [enablecompletion] => 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="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="showgrades">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="lang">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="enablecompletion">
                <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

moodle_enrol_manual_enrol_users

DEPRECATED: this deprecated function will be removed in a future version. This function has be renamed as enrol_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

moodle_file_get_files

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_files_get_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>
```

Error message

```
REST

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

Restricted to logged-in users
Yes

Callable from AJAX
No

moodle_file_upload

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_files_upload()

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

moodle_group_add_groupmembers

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_add_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

moodle_group_create_groups

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_create_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
}
)
```

```
XML-RPC (PHP structure)

[groups] =>
    Array
        (
        [0] =>
            Array
                (
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => string
                )
        )
```

REST (POST parameters)

```
groups[0][courseid]= int
groups[0][name]= string
groups[0][description]= string
groups[0][descriptionformat]= int
groups[0][enrolmentkey]= string
```

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 enrolm
ent 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 = MARKD
OWN)
enrolmentkey string   //group enrol secret phrase
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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

moodle_group_delete_groupmembers

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_delete_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

moodle_group_delete_groups

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_delete_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

moodle_group_get_course_groups

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_get_course_groups()

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   //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
}
)
```

XML-RPC (PHP structure)

```
    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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

moodle_group_get_groupmembers

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_get_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

moodle_group_get_groups

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_group_get_groups()

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   //multilang compatible name, course unique
description string   //group description text
descriptionformat int   //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARKD
OWN)
enrolmentkey string   //group enrol secret phrase
}
)
```

XML-RPC (PHP structure)

```


    Array
        (
        [0] =>
            Array
                (
                [id] => int
                [courseid] => int
                [name] => string
                [description] => string
                [descriptionformat] => int
                [enrolmentkey] => 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="enrolmentkey">
                <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

moodle_message_send_instantmessages

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_message_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 "1" //text format (1 = HTML, 0 = MOODLE, 2 = PLAIN or 4 = MARK
DOWN)
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
}
)
```

```
XML-RPC (PHP structure)


    Array
        (
        [0] =>
            Array
                (
                [msgid] => int
                [clientmsgid] => string
                [errormessage] => string
                )
        )
```

```
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>
        </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

moodle_notes_create_notes

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_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
No

moodle_role_assign

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_role_assign_role()

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
                                    (block, course, coursecat, system, user, module)
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

moodle_role_unassign

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_role_unassign_role()

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

moodle_user_create_users

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_user_create_users()

Arguments
users (Required)

General structure

```
list of (
object {
username string   //Username policy is defined in Moodle security config.
password string   //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
auth string  Default to "manual" //Auth plugins include manual, ldap, imap, etc
idnumber string  Default to "" //An arbitrary ID code number perhaps from the institution
lang string  Default to "de" //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
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, no HTML
city string  Optional //Home city of the user
country string  Optional //Home country code of the user, such as AU or CZ
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
preferences  Optional //User preferences
list of (
object {
type string   //The name of the preference
value string   //The value of the preference
}
)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
}
)}
)
```

XML-RPC (PHP structure)

```
[users] =>
    Array
        (
        [0] =>
            Array
                (
                [username] => string
                [password] => string
                [firstname] => string
                [lastname] => string
                [email] => string
                [auth] => string
                [idnumber] => string
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [city] => string
                [country] => string
                [firstnamephonetic] => string
                [lastnamephonetic] => string
                [middlename] => string
                [alternatename] => string
                [preferences] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [type] => string
                                [value] => string
                                )
                        )
                [customfields] =>
                    Array
                        (
                        [0] =>
                            Array
                                (
                                [type] => string
                                [value] => string
                                )
                        )
                )
        )
```

REST (POST parameters)

```
users[0][username]= string
users[0][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][auth]= string
users[0][idnumber]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][timezone]= string
users[0][mailformat]= int
users[0][description]= string
users[0][city]= string
users[0][country]= string
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
users[0][preferences][0][type]= string
users[0][preferences][0][value]= string
users[0][customfields][0][type]= string
users[0][customfields][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

moodle_user_delete_users

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_user_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

moodle_user_get_course_participants_by_id

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_user_get_course_user_profiles()

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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or
4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
name string   //The name of the custom field
shortname string   //The shortname of the custom field - to be able to build the field clas
s in the code
}
)preferences  Optional //Users preferences
list of (
object {
name string   //The name of the preferences
value string   //The value of the custom field
}
)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
}
)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
                (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [icq] => string
                [skype] => string
                [yahoo] => string
                [aim] => string
                [msn] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [auth] => string
                [confirmed] => int
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [url] => string
                [country] => string
                [profileimageurlsmall] => string
                [profileimageurl] => string
                [customfields] =>
                    Array
                    (
                    [0] =>
                        Array
                            (
                            [type] => string
                            [value] => 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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="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="url">
                <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="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

moodle_user_get_users_by_courseid

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_enrol_get_enrolled_users()

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. * userfields ('string, string, ...') return only the values of these user fields. * limitfrom (integer) sql limit from. * limitnumber (integer) maximum number of returned users.

```
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 c
ourse has groups enabled and this param
                                                isn't defined, returns all the viewable user
s.
                                                This option requires 'moodle/site:accessallg
roups' 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 cour
se context.
                            * userfields ('string, string, ...') return only the values of t
hese user fields.
                            * limitfrom (integer) sql limit from.
                            * limitnumber (integer) maximum number of returned users.
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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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)
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
url string  Optional //URL 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 clas
s 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
}
)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
                (
                [id] => int
                [username] => string
                [firstname] => string
                [lastname] => string
                [fullname] => string
                [email] => string
                [address] => string
                [phone1] => string
                [phone2] => string
                [icq] => string
                [skype] => string
                [yahoo] => string
                [aim] => string
                [msn] => string
                [department] => string
                [institution] => string
                [idnumber] => string
                [interests] => string
                [firstaccess] => int
                [lastaccess] => int
                [description] => string
                [descriptionformat] => int
                [city] => string
                [url] => 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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="description">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="descriptionformat">
                <VALUE>int</VALUE>
            </KEY>
            <KEY name="city">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="url">
                <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>
            <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

moodle_user_get_users_by_id

DEPRECATED: this deprecated function will be removed in a future version. Use core_user_get_users_by_field service instead

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

```
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
icq string  Optional //icq number
skype string  Optional //skype id
yahoo string  Optional //yahoo id
aim string  Optional //aim id
msn string  Optional //msn number
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, imap, etc
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 //description format (1 = HTML, 0 = MOODLE, 2 = PLAIN or
4 = MARKDOWN)
city string  Optional //Home city of the user
url string  Optional //URL 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
name string   //The name of the custom field
shortname string   //The shortname of the custom field - to be able to build the field clas
s in the code
}
)preferences  Optional //Users 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 course
s 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
            [icq] => string
            [skype] => string
            [yahoo] => string
            [aim] => string
            [msn] => string
            [department] => string
            [institution] => string
            [idnumber] => string
            [interests] => string
            [firstaccess] => int
            [lastaccess] => int
            [auth] => string
            [confirmed] => int
            [lang] => string
            [calendartype] => string
            [theme] => string
            [timezone] => string
            [mailformat] => int
            [description] => string
            [descriptionformat] => int
            [city] => string
            [url] => string
            [country] => string
            [profileimageurlsmall] => string
            [profileimageurl] => string
            [customfields] =>
                Array
                    (
                    [0] =>
                        Array
                            (
                            [type] => string
                            [value] => string
                            [name] => string
                            [shortname] => string
                            )
                    )
            [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="icq">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="skype">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="yahoo">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="aim">
                <VALUE>string</VALUE>
            </KEY>
            <KEY name="msn">
                <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="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="url">
                <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="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

moodle_user_update_users

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_user_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.
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
auth string  Optional //Auth plugins include manual, ldap, imap, etc
idnumber string  Optional //An arbitrary ID code number perhaps from the institution
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, no HTML
city string  Optional //Home city of the user
country string  Optional //Home country code of the user, such as AU or CZ
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
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
                [password] => string
                [firstname] => string
                [lastname] => string
                [email] => string
                [auth] => string
                [idnumber] => string
                [lang] => string
                [calendartype] => string
                [theme] => string
                [timezone] => string
                [mailformat] => int
                [description] => string
                [city] => string
                [country] => string
                [firstnamephonetic] => string
                [lastnamephonetic] => string
                [middlename] => string
                [alternatename] => string
                [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][password]= string
users[0][firstname]= string
users[0][lastname]= string
users[0][email]= string
users[0][auth]= string
users[0][idnumber]= string
users[0][lang]= string
users[0][calendartype]= string
users[0][theme]= string
users[0][timezone]= string
users[0][mailformat]= int
users[0][description]= string
users[0][city]= string
users[0][country]= string
users[0][firstnamephonetic]= string
users[0][lastnamephonetic]= string
users[0][middlename]= string
users[0][alternatename]= string
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

Error message

```
REST

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

Restricted to logged-in users
Yes

Callable from AJAX
No

moodle_webservice_get_siteinfo

DEPRECATED: this deprecated function will be removed in a future version. This function has been renamed as core_webservice_get_site_info()

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   //user 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 forcelogin 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
}
```

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
        )
```

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

Error message

```
REST

<?xml version="1.0" encoding="UTF-8"?>
<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_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
```

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
Yes

Callable from AJAX
Yes

tool_templatelibrary_load_canonical_template

Load a canonical template by name (not the theme overidden 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

Yes

Callable from AJAX
Yes

Moodle Docs for this page

You are logged in as Admin Nutzer (Log out)
Home