

CampusNet public API

1. Introduction	4
2. Using the API	4
2.1. Authentication and authorization	4
2.2. Obtaining the GUID for users	4
2.3. Obtaining the GUID for groups	4
2.4. Obtaining a protected access password	4
2.5. Obtaining an app token	5
2.6. Data formats	6
2.7. Errors	6
2.8. Integrating with Dropbox	6
3. Resources using GUID authentication	7
3.1. Element/{elementGuid}/Files/Newest	7
3.2. Element/{elementGuid}/Messages/Newest	7
3.3. User/{userGuid}/Calendar/Upcoming	7
3.4. User/{userGuid}/Files/Newest	7
3.5. User/{userGuid}/Messages/Newest	7
4. Resources using protected access password authentication	8
4.1. CurrentUser/Cafeteria	8
4.2. CurrentUser/Calendar/{startDate}/To/{endDate}	9
4.3. CurrentUser/Calendar/FeedInfo	9
4.4. CurrentUser/Calendar/Search?query={query}&first={first}&amount={amount}	9
4.5. CurrentUser/Calendar/Upcoming	10
4.6. CurrentUser/Calendar/Upcoming/From/{first}/Returning/{amount}	10
4.7. CurrentUser/Dropbox/RequestToken (POST)	10
4.8. CurrentUser/Dropbox/Authorization (GET)	11
4.9. CurrentUser/Dropbox/Authorization (POST)	11
4.10. CurrentUser/Dropbox/Authorization (DELETE)	12
4.11. CurrentUser/Dropbox/Upload (POST)	12
4.12. CurrentUser/Elements	13
4.13. CurrentUser/Elements/{elementId}	14
4.14. CurrentUser/Elements/{elementId}/Calendar/{StartDate}/To/{EndDate}	14
4.15. CurrentUser/Elements/{elementId}/Calendar/FeedInfo	14
4.16. CurrentUser/Elements/{elementId}/Calendar/Upcoming/From/{first}/Returning/{amount}	15
4.17. CurrentUser/Elements/{elementId}/Elements	15
4.18. CurrentUser/Elements/{elementId}/Files	16
4.19. CurrentUser/Elements/{elementId}/Files/{fileId}/Bytes	17
4.20. CurrentUser/Elements/{elementId}/Files/{fileId}/Versions/{version}/Bytes	17
4.21. CurrentUser/Elements/{elementId}/Files/Newest/From/{first}/Returning/{amount}	17
4.22. CurrentUser/Elements/{elementId}/Files/Upload?folderId={folderId} (POST)	17
4.23. CurrentUser/Elements/{elementId}/FrontPage	18

4.23.1. accept: text/html	18
4.23.2. accept: text/plain	18
4.24. CurrentUser/Elements/{elementId}/Links	18
4.25. CurrentUser/Elements/{elementId}/Messages (POST).....	19
4.26. CurrentUser/Elements/{elementId}/Messages/Newest.....	19
4.27. CurrentUser/Elements/{elementId}/Messages/Newest/From/{first}/Returning/{amount}	19
4.28. CurrentUser/Elements/{elementId}/Participants	20
4.29. CurrentUser/Elements/Search?query={query}&first={first}&amount={amount}	20
4.30. CurrentUser/Files	21
4.31. CurrentUser/Files/{fileId}/Bytes.....	21
4.32. CurrentUser/Files/{fileId}/Versions/{version}/Bytes.....	21
4.33. CurrentUser/Files/Newest.....	21
4.34. CurrentUser/Files/Newest/From/{first}/Returning/{amount}.....	21
4.35. CurrentUser/Files/Search?query={query}&first={first}&amount={amount}	21
4.36. CurrentUser/Files/Upload?folderId={folderId} (POST)	22
4.37. CurrentUser/Grades	22
4.38. CurrentUser/InformationSources.....	23
4.39. CurrentUser/InformationSources/{id}.....	23
4.40. CurrentUser/Library/Feed	24
4.41. CurrentUser/Library/Holds.....	24
4.42. CurrentUser/Library/Information.....	25
4.43. CurrentUser/Library/Loans.....	25
4.44. CurrentUser/Library/Requests	26
4.45. CurrentUser/Links.....	26
4.46. CurrentUser/Messages/Newest	26
4.46.1. accept: text/html	27
4.46.2. accept: text/plain	27
4.46.3. accept: application/atom+xml	27
4.47. CurrentUser/Messages/Newest/From/{first}/Returning/{amount}	27
4.48. CurrentUser/Messages/Search?query={query}&first={first}&amount={amount}.....	27
4.49. CurrentUser/UserInfo.....	28
4.49.1. accept: application/xml.....	28
4.49.2. accept: text/x-vcard.....	28
4.50. CurrentUser/Users/{userId}	29
4.51. CurrentUser/Users/{userId}/Picture	29
4.52. CurrentUser/Users/{userId}/Picture/Width/{width}.....	29
4.53. CurrentUser/Users/Search?query={query}&first={first}&amount={amount}	29
4.54. CurrentUser/Polls/From/{first}/To/{amount}	31
4.55. CurrentUser/Polls/{id}.....	32
4.56. CurrentUser/Polls/{id}/Results.....	32
4.57. CurrentUser/Polls/{id}/Vote (POST)	33
4.58. CurrentUser/Events.....	34
4.59. CurrentUser/Events/Recommended.....	35
4.60. CurrentUser/Events/Archived	35

5. Beta functions.....	35
5.1. CurrrentUser/Elements/{elementId}/Messages/v2 (GET)	35
5.2. CurrrentUser/Elements/{elementId}/Messages/v2/From/{first}/returning/{amount}/ (GET).....	36
5.3. CurrrentUser/Elements/{elementId}/Messages/v2/{messageId} (GET)	36
5.4. CurrentUser/Elements/{elementId}/Messages/v2 (POST)	37
5.5. CurrentUser/Elements/{elementId}/Messages/v2/{messageId} (PUT)	38
5.6. CurrentUser/Elements/{elementId}/Messages/v2/{messageId} (DELETE)	38
5.7. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments (GET)	38
5.8. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments/{commentId} (DELETE)	39
5.9. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments (POST)	39
5.10. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments/{commentId}(PUT)	40
5.11. CurrentUser/Elements/{elementId}/Messages/v2/NotificationSettings (GET)	40
5.12. CurrentUser/Elements/{elementId}/Messages/v2/NotificationSettings (PUT).....	41
5.13. CurrentUser/Elements/{elementId}/Messages/v2/{messageId }/NotificationSettings (GET)	42
5.14. CurrentUser/Elements/{elementId}/Messages/v2/{messageId }/NotificationSettings (PUT).....	42
5.15. CurrentUser/Messages/v2/{messageId }/RegisterView (PUT).....	43

1. Introduction

CampusNet contains a RESTful HTTP API for extracting data. The base address is

<https://www.campusnet.<university>.dk/data/>

The API is currently in development and may change without warning. Additional features will be added along the way.

2. Using the API

2.1. Authentication and authorization

The API contains three types of authentication:

1. To each resource (group or user) we associate a GUID, giving read-only access to the most basic functionality such as a list of the newest messages and files, such as you see on the CampusNet front page. If you know the GUID, i.e. the URL, you can read the data. The primary purpose is to allow access to this data from feed readers.
2. Each user is associated a “protected access” password to his account. This password currently has GUID format which may cause confusion. The username is his usual username used for CampusNet login. This gives access to more content than the user GUID, such as downloading files from the CampusNet file sharing, getting a list of his groups and so on. It does not give access to other systems such as course enrollment and so on. The reason we don’t just use the CAS password for authentication is mainly that we don’t want users’ CAS passwords saved in apps – it grants access to too many systems.

Furthermore an authentication token is created for each App, prior to extracting data. This is a requirement in order to ensure that API calls from an App can be closed if misused.

3. The token is created when the App is registered, and sent to the provided email address. Clients (Apps) will need to provide the App name, as well as the generated token, each time an API call is made.

2.2. Obtaining the GUID for users

Currently the user GUID cannot be obtained anywhere. This feature is planned.

2.3. Obtaining the GUID for groups

The group GUID is available on the group front-page using the RSS icons. These icons only show up if content is marked public in the group (Admin -> Edit group name and properties). Similarly, even if you know the GUID, access is only granted if content is marked public.

2.4. Obtaining a protected access password

To obtain the protected access password, perform these steps:

Ask the user for his CAS username and password, the one used for normal CampusNet login.

Create a HTTPS post containing the username and password. The URL is your CAS URL followed by mobilapp.jsp, for example <https://auth.dtu.dk/dtu/mobilapp.jsp>.

Example:

```
POST https://auth.dtu.dk/dtu/mobilapp.jsp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: auth.dtu.dk
Content-Length: 31

username=abcd&password=12345678
```

The page will return XML containing the protected access password in this format:

```
<xml>
  <LimitedAccess Password="<string containing the protected access password>" />
</xml>
```

Example:

```
<xml>
  <LimitedAccess Password="21EF8196-ED05-4BAB-9081-44313ABD3D32" />
</xml>
```

The limited access password will normally not change. In the future perhaps the user will be able to reset it. We recommend locally saving the username and protected access password for future logons.

If the user enters a wrong password his IP address will be blocked from trying again for a little while. This is to safeguard against brute force attacks. When this happens, CAS will return the following:

```
<xml>
  <BlockedAccess Ip="<IP address of the user>" Reason="<string containing the reason for
being blocked. This is currently always 'IpWrongUserCredentials
'>" TryAgainIn="<duration specifying how long you have to wait before being allowed to try
again>" />
</xml>
```

The format for TryAgainIn is [d.]hh:mm:ss[.ffffff] where d is the number of days (24 hour periods), hh is the number of hours, mm is the number of minutes, ss is the number of seconds and fffffff represent fractional seconds.

Example:

```
<xml>
  <BlockedAccess Ip="130.225.95.254" Reason="IpWrongUserCredentials"
TryAgainIn="00:00:04.2656297" />
</xml>
```

2.5. Obtaining an app token

To obtain the token needed for the App client calls, perform these steps:

Visit the App registration site:

<https://www.campusnet.<university>.dk/data/Documentation/RequestApiCredentials.aspx>

- Enter the name of your App (must be unique).
- Enter a contact name.
- Enter a valid email address (the token will be sent to this email).

2.6. Data formats

All requests are GET unless otherwise mentioned.

Most data is returned in XML. For each method the format is described and there may be an example as well. All dates and times are CET.

Some data is returned in Atom 1.0 format plus some extra fields contained in the <http://campusnet.arcenic.dk/> XML namespace, here called `cn`. This is what is meant below when referring to the Atom format. All our Atom formats contain the following extra fields:

`entry/cn:element`, containing the name of a group plus the CampusNet URL for it. If it is a personal item, this entry is omitted.

`entry/cn:service`, containing the name the CampusNet service to which the data belongs.

`entry/cn:type`, containing a code for the CampusNet service to which the data belongs.

2.7. Errors

In case of bad user input, error codes will be returned. You will get an informative HTTP status code, plus XML content indicating an error name and description, such as this example:

```
<Error xmlns="http://schemas.datacontract.org/2004/07/Arcenic.CampusNet.DataService"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Description>No access to specified element</Description>
  <Name>NoAccess</Name>
</Error>
```

2.8. Integrating with Dropbox

If you want to transfer files from CampusNet to Dropbox you have two options:

1. Download the file and then use the Dropbox API to upload the file.
2. Transfer the file directly from the CampusNet file servers.

The second option is of course much nicer for users on slow connections. However, it is slightly more complicated to implement, partly because of these constraints built into the Dropbox API:

- User access tokens cannot be transferred between apps having different app credentials.
- Dropbox does not allow saving passwords, only access tokens.
- Authorization must use OAuth.

As such, CampusNet must authorize with the user's Dropbox account. The CampusNet API uses OAuth 1.0. The process for Dropbox authorization is:

1. Request a token from the CampusNet API. (4.7)

2. Get the user to authorize CampusNet using the following URL where {request token} is the token you requested in step 1.

```
https://www.dropbox.com/1/oauth/authorize?oauth_token={request token}
```

3. Authorize the token with the CampusNet API. (4.9)

It is also possible to delete the credentials stored in the CampusNet database.

3. Resources using GUID authentication

These resources rely on the correct `userGuid` or `elementGuid` being specified. Possible errors returned from these resources are

- 400, `BadGuidFormat` if the GUID specified could not be parsed as a GUID.
- 404, `UserNotFound` if no active user exists, having the specified GUID.
- 404, `ElementNotFound` if no element exists, having the specified GUID.
- 403, `NoAccess` if element exists but content is not marked public.

3.1. `Element/{elementGuid}/Files/Newest`

Returns info about 10 newest files from the specified group.

Format: Atom.

3.2. `Element/{elementGuid}/Messages/Newest`

Returns 10 newest messages from the specified group.

Format: Atom.

3.3. `User/{userGuid}/Calendar/Upcoming`

Returns next 10 calendar appointments from integrated groups.

Format:

Atom having these extra fields:

`entry/cn:duration`, containing the attributes `StartTime` and `EndTime` indicating the duration of the appointment.

3.4. `User/{userGuid}/Files/Newest`

Returns info about 10 newest files from integrated groups.

Format: Atom.

3.5. `User/{userGuid}/Messages/Newest`

Returns 10 newest messages from integrated groups.

Format: Atom.

4. Resources using protected access password authentication

These resources use HTTP basic authentication using the protected access password discussed in 2.1. This means adding a HTTP header containing the username and protected access password.

These resources also need you to specify app name and app token in the request. They are specified in the HTTP headers `X-appname` and `X-token` respectively.

Example request:

```
GET https://www.campusnet.dtu.dk/data/CurrentUser/UserInfo HTTP/1.1
Host: www.campusnet.dtu.dk
Authorization: Basic dGVzdHVzZXI6Mzk0NmEzODktYjZjNS00Zjc3LWJlYWMTMzg0NjA1YjBkMmMz
X-appname: My first app
X-token: 2c62c0aa-e403-49ec-bd4f-0b774f5a4931
```

A few of the resources/methods additionally accept URL query string parameters for authentication. This allows sending the URL to a browser for downloading files. The parameter names are `serviceUsername`, specifying your username, `servicePassword`, specifying your protected access password, `appName`, specifying the name registered for your app, and `appToken` for the token that belongs to your app. Methods that support this kind of authentication mention it in the documentation below.

The resources taking an `elementId` input require the `elementId` to be an integer, specifying which CampusNet element to handle. Possible errors are:

- 400, `BadElementIdFormat` if the `elementId` could not be parsed as an integer.
- 403, `NoAccess` if the user has no access to the specified element.

Element services (messages, calendar, files, participants) may be disabled by a group administrator. If you attempt accessing a disabled service, you will get

- 403, `ServiceNotEnabled`

4.1. CurrentUser/Cafeteria

Return all cafeterias with the menu of the current week.

Format:

```
<Cafeterias>
  <Cafeteria Id="<Guid of the cafeteria>" Link="<url to the cafeterias homepage>"
Name="<Name of the cafeteria>">
  <Menus Date="<Date when the menus are active >" EndDate="<If the menus are active more
than one day, this indicates the last active day>">
    <Menu Title="<title of the menu >" Text="<Description of the menu>" Price="<Price of
the menu>" />
    ...
  </Menus>
</Cafeteria>
  ...
</Cafeterias>
```


Example:

```
<Cafeterias>
  <Cafeteria Id="4d410f66-5c7e-49d7-b17f-ad71d09623c8"
Link="http://www.facultyclub.com/restaurant" Name="Faculty Club, Restaurant">
  <Menus Date="25-08-2014" EndDate="31-08-2014">
    <Menu Title="Ugens fisk" Text="Filleteret torsk, pyntet med kapers og citron"
Price="34"/>
    <Menu Title="Ugens ret" Text="Oksekøds lasagne med parmesan" Price="20"/>
  </Menus>
</Cafeteria>
</Cafeterias>
```

4.2. CurrentUser/Calendar/{startDate}/To/{endDate}

Returns calendar appointments that start on a date between `startDate` and `endDate`, inclusive. Personal appointments and appointments from integrated groups are both included.

Errors:

- 400, `BadDateFormat` if `startDate` or `endDate` could not be parsed.

Format: see 3.3.

4.3. CurrentUser/Calendar/FeedInfo

Returns info about your personal calendar feed. It includes appointments from all your groups that have public content and from your personal calendar.

Format:

```
<Calendar>
  <ICal Url="<Url that points to an iCal feed>" />
</Calendar>
```

Example:

```
<Calendar>
  <ICal Url="https://www.campusnet.dtu.dk/rss/CampusNetCalendar.asp?userguid= E16A6512-848F-4210-9424-1E422EC5CC0C&lang=en" />
</Calendar>
```

4.4. CurrentUser/Calendar/Search?query={query}&first={first}&amount={amount}

Performs a search among calendar entries in all your active groups in the CampusNet database using `query` as the search string. The search algorithm supports double quotes for exact matches and the exclusion operator `"-"`. Results are sorted based on ranking, not based on dates.

The parameters `first` and `amount` control which portion of the search results are shown. These are optional and default to 1 and 20, respectively. The maximum value for both values is 100.

If more parameters are added to the specification at some point, they will be optional as well.

Format: Atom

Example URL: `CurrentUser/Calendar/Search?query=meeting&first=1&amount=10`

Errors:

- 400, `NoQuery` if `query` is empty or not specified at all.
- 400, `FormatException` if `first` or `amount` is specified but could not be parsed as an integer.
- 400, `ArgumentException` if `first` or `amount` is specified but is not positive.
- 400, `ArgumentException` if `first` or `amount` is specified but is higher than the maximum allowed value.
- 400, `BadQuery` if the search could not be completed. This is normally due to all search terms being ignored for one reason or another.
- 504, `Timeout` if performing the search took too long. This doesn't seem to happen currently, but may happen later if the search algorithm is changed and searches for very short string are performed.

4.5. `CurrentUser/Calendar/Upcoming`

Deprecated – use `CurrentUser/Calendar/Upcoming/From/{first}/Returning/{amount}` instead.

Returns next 10 calendar appointments from integrated groups.

Format: see 3.3.

4.6. `CurrentUser/Calendar/Upcoming/From/{first}/Returning/{amount}`

Returns `amount` next calendar appointments from integrated groups, starting from `first`.

Errors:

- 400, `FormatException` if `first` or `amount` could not be parsed as integers.
- 400, `ArgumentException` if `first` or `amount` are not positive.

Format: see 3.3.

4.7. `CurrentUser/Dropbox/RequestToken (POST)`

Gets a request token from Dropbox that the user must authorize at the following URL where `{request token}` is the token returned from this API method. See the Dropbox Core API documentation for more information on what parameters you can use with this URL.

`https://www.dropbox.com/1/oauth/authorize?oauth_token={request token}`

Example:

```
POST /data/CurrentUser/Dropbox/RequestToken HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Authorization: Basic blabla
```

Format:

```
<Token>{request token}</Token>
```

Errors:

- XXX, `DropboxException` if the CN API experiences an error using the Dropbox API. The status code from Dropbox is forwarded to the user of the CN API.

4.8. CurrentUser/Dropbox/Authorization (GET)

Gets the username for the current Dropbox authorization, if one exists.

Example:

```
GET /data/CurrentUser/Dropbox/Authorization HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Length: 0
Authorization: Basic blabla
```

Format:

```
<Authorized>{dropbox username}</Authorized>
```

Or:

```
<NotAuthorized/>
```

4.9. CurrentUser/Dropbox/Authorization (POST)

Uses the token (returned from the call to 4.74.7) to get an access token from Dropbox. The user must have authorized the token. The username and access token are then stored in the database. If successful, the dropbox username is returned. Post data (application/x-www-urlform-encoded):

- `token`: The token received from calling 4.7.

Arguments must be placed in the body of the request, being application/x-www-form-urlencoded.

Example :

```
POST /data/CurrentUser/Dropbox/Authorization HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Authorization: Basic blabla
```

```
token=yourToken
```

Format:

```
<Success>{dropbox username}</Success>
```

Errors:

- 400, `ArgumentException` if `token` is not specified.
- XXX, `DropboxException` if the CN API experiences an error using the Dropbox API. The status code from Dropbox is forwarded to the user of the CN API. This could happen if the password is wrong.

4.10. CurrentUser/Dropbox/Authorization (DELETE)

Deletes the Dropbox authorization stored for the user, if any. It just returns status code 200 and no content.

Example:

```
DELETE /data/CurrentUser/Dropbox/Authorization HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Length: 0
Authorization: Basic blabla
```

4.11. CurrentUser/Dropbox/Upload (POST)

Uses the Dropbox access token stored in the database to transfer a file from the CN files sharing to Dropbox. Post data (application/x-www-form-urlencoded):

- `userName`: The user name for the Dropbox account. This is to ensure that you don't upload to the wrong account if there are different Dropbox credentials stored for your CampusNet user.
- `path`: The path to upload to in Dropbox. Always starts with `"/`.
- `elementId`: If the file you are uploading belongs to a group in CampusNet, specify the element ID here, otherwise omit it completely or leave it blank.
- `fileId`: ID of the file to upload.
- `version`: Version of the file to upload. If omitted, the newest version is automatically chosen.

Example:

```
POST /data/CurrentUser/Dropbox/Upload HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Type: application/x-www-form-urlencoded
Content-Length: 104
Authorization: Basic blabla
```

```
userName=username%40example.com&elementId=3063&path=%2FMy+folder%2FMy+subfolder&fileId=1389813
&version=4
```

Errors:

- 400, `FormatException` if `fileId` or `version` could not be parsed as integers.
- 400, `ArgumentException` if `userName`, `path` or `fileId` are not specified.
- 403, `CredentialsNotSet` if the current user has not saved his Dropbox credentials or if the saved credentials belong to another username than specified in `userName`.
- XXX, `DropboxException` if the CN API experiences an error using the Dropbox API. The status code from Dropbox is forwarded to the user of the CN API. This could happen if the stored password is wrong or the disk quota is exceeded.

4.12. CurrentUser/Elements

Returns a list of the groups seen in the menu on the CampusNet front page. They are grouped and in the same order as on the front page. Only groupings containing elements are returned.

Normally only group information is returned. If you want information about the user's rights and enabled services, include the header `X-Include-services-and-relations` and set it to `true`.

Format without services and relations:

```
<ElementGroupings>
  <Grouping Id="Name of the grouping">
    <Element Id="Unique int32 id for the element" ParentId="<int32 id of parent element. 0
if none>" Name="Name of the element" ContextName="Context name of the element. The
context name includes names of parent elements" IsArchived="Boolean ('true' or 'false'
indicating whether the group was activated through the archive)" SubgroupCount="<int32
number of subgroups accessible via the menu"/>
    ..
  </Grouping>
  ..
</ElementGroupings>
```

Format with services and relations:

```
<ElementGroupings>
  <Grouping Id="Name of the grouping">
    <Element <Same format as above>
      <UserElementRelation <Same format as in 4.13>/>
      <Services <Same format as in 4.13>/>
    </Element/>
    ..
  </Grouping>
  ..
</ElementGroupings>
```

Both Danish and English names and context names can be returned. Use the `accept-language` header to specify either `da-DK` or `en-GB`. English is default.

If `IsArchived` is true we recommend that your user interface encloses the name of the group in square brackets [], such that the user experience is similar to that of the web edition of CampusNet.

Example:

```
<ElementGroupings>
  <Grouping Id="Groups">
    <Element Id="228807" ParentId="20219" Name="MAT1TOP10" ContextName="Institut for
Matematik/MAT1TOP10" IsArchived="false" SubgroupCount="0"/>
  </Grouping>
  <Grouping Id="User groups">
    <Element Id="35673" ParentId="0" Name="Bordfodbold" ContextName="Bordfodbold"
IsArchived="true" SubgroupCount="0"/>
    <Element Id="249683" ParentId="20197" Name="Den seje gruppe" ContextName="The cool
group" IsArchived="false" SubgroupCount="2"/>
  </Grouping>
</ElementGroupings>
```

4.13. CurrentUser/Elements/{elementId}

Returns basic information about the group.

Format:

```
<Element Id="Unique int32 id for the element" ParentId="int32 id of parent element. 0 if none." Name="Name of the element" ContextName="Context name of the element. The context name includes names of parent elements" IsArchived="Boolean ('true' or 'false') indicating whether the group was activated through the archive" SubgroupCount="int32 number of subgroups accessible via the menu"/>
  <UserElementRelation ACL="Acl is described below">
    <Services>
      <Service Uid="Id of a service that is enabled on the group and to which you have access. Only ids of services that can be accessed through this API are returned"
SmsAllowedForGroup="Boolean that is returned for the Messages service. It tells you whether the group allows authors and administrators to send SMS" HomepageUrl ="String that is returned for the Homepage service, telling you the URL of the homepage. If the homepage service is active, this value is always present."/>
      ...
    </Services>
  </Element>
```

ACL: Access control list is a name identifying the rights a user has on the group. Possible values are `Administrator`, `Author` and `User`. Generally a user having `ACL = User` cannot edit anything. The two other levels can edit element data.

Both Danish and English names and context names of groups can be returned. Use the `accept-language` header to specify either `da-DK` or `en-GB`. English is default.

Example:

```
<Element Id="207935" ParentId="0" Name="test" ContextName="test" IsArchived="false" SubgroupCount="3">
  <UserElementRelation ACL="Administrator">
    <Services>
      <Service Uid="Calendar"/>
      <Service Uid="Files"/>
      <Service Uid="Homepage" HomepageUrl="http://duckduckgo.com"/>
      <Service Uid="Messages" SmsAllowedForGroup="true"/>
      <Service Uid="Participants"/>
    </Services>
  </Element>
```

4.14. CurrentUser/Elements/{elementId}/Calendar/{StartDate}/To/{EndDate}

Returns calendar appointments from the specified element that start on a date between `startDate` and `endDate`, inclusive.

Errors: see 4.2.

Format: see 3.3.

4.15. CurrentUser/Elements/{elementId}/Calendar/FeedInfo

Returns info about the calendar feed belonging to the group.

Format:

```
<Calendar FeedsAllowed="<Boolean ('True' or 'False') indicating whether the calendar is
available as a feed. This can be turned on or off by a group administrator.">
  <ICal Url="<Url that points to an iCal feed>" />
</Calendar>
```

The ICal element is only present if feeds are allowed.

Example 1:

```
<Calendar FeedsAllowed="True">
  <ICal Url="https://www.campusnet.dtu.dk/rss/CampusNetCalendar.asp?elementguid=E16A6512-
848F-4210-9424-1E422EC5CC0C&lang=en" />
</Calendar>
```

Example 2:

```
<Calendar FeedsAllowed="False" />
```

4.16. CurrentUser/Elements/{elementId}/Calendar/Upcoming/From/{first}/Returning/{amount}

Returns `amount` next calendar appointments from specified group, starting from `first`.

Errors:

- 400, `FormatException` if `first` or `amount` could not be parsed as integers.
- 400, `ArgumentException` if `first` or `amount` are not positive.

Format: see 3.3.

4.17. CurrentUser/Elements/{elementId}/Elements

Returns a list of subgroups for the specified group. They are the same as the ones shown in the CampusNet menu.

Normally only group information is returned. If you want information about the user's rights and enabled services, include the header `X-Include-services-and-relations` and set it to `true`.

Both Danish and English names and context names can be returned. Use the `accept-language` header to specify either `da-DK` or `en-GB`. English is default.

Format without services and relations:

```
<Elements>
  <Element Id="<Unique int32 id for the element>" ParentId="<int32 id of parent element. 0
if none>" Name="<Name of the element>" ContextName="<Context name of the element. The
context name includes names of parent elements>" IsArchived="<Boolean ('True' or 'False')
indicating whether the group was activated through the archive>" SubgroupCount="<int32
number of subgroups accessible via the menu>" />
  ..
</Elements>
```

Format with services and relations:

```

<Elements>
  <Element <Same format as above>>
    <UserElementRelation <Same format as in 4.13>/>
    <Services <Same format as in 4.13>/>
  <Element/>
  ..
</Elements>

```

Example:

```

<Elements>
  <Element Id="35673" ParentId="123" Name="Bordfodbold" ContextName="Bordfodbold"
IsArchived="true" SubgroupCount="0"/>
  <Element Id="249683" ParentId="123" Name="Den seje gruppe" ContextName="The cool group"
IsArchived="false" SubgroupCount="2"/>
</Elements>

```

4.18. CurrentUser/Elements/{elementId}/Files

Returns all files in the file sharing for the specified group.

Format:

```

<Folder Id="<int64 unique id of the folder>" Name="<Name of the folder>" Created="<ISO date
designating when the folder was created>" Note="<Text describing the folder>"
MayUpload="<Boolean ('True' or 'False') indicating whether the current user is allowed to
upload to the folder>">
  <Folder ... />
  ...
  <File Id="<int64 unique id of the file>" Name="<Name of the file>">
    <FileVersion Version="<int32 version of the file>" KbSize="<Size of the file in Kb>"
Created="<ISO date designating when the folder was created>" CreatorId="<int32 id of the
user who uploaded the file >"
Note="<Text describing the file>"/>
    ...
  </File>
</Folder>

```

The hierarchical structure of the file sharing folders and files are reflected in the format. A root folder is always returned. Each folder can contain folders and files. Each file contains at least one version. The folders and files are ordered the same way as when viewed through the CampusNet website, i.e. sometimes using user defined ordering.

If there is no note for a folder or file version, the attribute is omitted to save bandwidth. The version numbering starts at 1 but if a new version exists and an old version of the same file was deleted, there will be a “hole” in the numbering. If a file is unavailable to you, perhaps due to availability dates, the file will not be returned.

By providing an `accept` header when requesting, you can choose between formats of the note attribute. The default format is `text/html` but `text/plain` is also available.

Example:

```

<Folder Id="131196" Name="Top" Created="2005-08-30T12:33:00.377" MayUpload="False">

```



```

    <Folder Id="511815" Name="Møde nr. 1 - 9. november 2005" Created="2010-05-05T13:35:22.28"
Note="Referater o.lign" MayUpload="True">
    <File Id="2466303" Name="Dagsorden.doc">
        <FileVersion Version="1" KbSize="96" Created="2005-11-22T08:17:21" CreatorId="50972"
Note="Dagsorden til mødet"/>
        <FileVersion Version="2" KbSize="118" Created="2005-11-22T08:18:16" CreatorId="50972"
Note="Dagsorden til mødet + noter"/>
    </File>
</Folder>
<File Id="2083622" Name="abc.txt">
    <FileVersion Version="1" KbSize="5" CreatorId="50972" Created="2009-04-30T08:39:58.86"/>
</File>
</Folder>

```

4.19. CurrentUser/Elements/{elementId}/Files/{fileId}/Bytes

Returns the file contents of a file belonging to the file sharing of a group, with a specific `fileId`.

This resource accepts the URL query string parameters `serviceUsername` and `servicePassword`.

Format: Raw bytes

The headers include `content-length`.

The file name is included in the `content-disposition` header. Note that it uses the extended notation of RFC 5987 in order to support Unicode.

4.20. CurrentUser/Elements/{elementId}/Files/{fileId}/Versions/{version}/Bytes

Returns the file contents of a specific version of a file as indicated by `fileId` and `version`.

This resource accepts the URL query string parameters `serviceUsername` and `servicePassword`.

Format: see 4.19.

4.21. CurrentUser/Elements/{elementId}/Files/Newest/From/{first}/Returning/{amount}

Returns info about `amount` newest files from specified group, starting from `first`.

Format: see 3.4.

4.22. CurrentUser/Elements/{elementId}/Files/Upload?folderId={folderId} (POST)

Uploads a file to the file sharing in the element and folder specified by `elementId` and `folderId`. If the file already exists, a new version will be uploaded. The file contents (bytes) are to be sent in the post data. The file name must be sent in a header called `X-File-Name`. This is compatible with the `send` method of the XMLHttpRequest2 standard and is therefore also possible to use from HTML5 with JavaScript. We don't support chunked upload. The service returns status 200 if the upload succeeded.

Example:

```
POST /data/CurrentUser/Elements/3063/Files/Upload?folderId=1234 HTTP/1.1
```

Host: www.campusnet.dtu.dk
Content-Length: 22
Authorization: Basic blabla
X-File-Name: file.txt
Content-Type: multipart/form-data

Her er filens indhold.

Errors:

- 400, `FormatException` if `folderId` is not an integer.
- 400, `MissingHeader` if `X-File-Name` is not specified.
- 403, `QuotaExceeded` if your uploaded files in user groups plus your personal file sharing exceeds the quota. Your quota and currently used storage can be seen in the web version of CampusNet.
- 403, `NoUploadRights` if you are not allowed to upload to the folder specified by `folderId`.
- 400, `InvalidFileName` if the file name is not valid in the file sharing. This could be due to length limitations or characters such as `/` or `\` being used.

4.23. `CurrentUser/Elements/{elementId}/FrontPage`

Returns the title and content of the front page for the specified group.

Format: By giving an `accept` header when requesting, you can choose between formats. The default format is `text/html`.

4.23.1. `accept: text/html`

`<ElementFrontpage Title="<Title shown on the front page of the group, using the language of current user>" Content="<HTML content of the frontpage>" />`

Example:

`<ElementFrontpage Title="Velkommen til kurset" Content="<p>Kurset foregårår i bygning 306.</p><p>Vi ses!</p>" />`

4.23.2. `accept: text/plain`

As `text/html`, but the HTML content is converted to plaintext before being returned.

Example:

`<ElementFrontpage Title="Velkommen til kurset" Content="Kurset foregår i bygning 306.
Vi ses!" />`

4.24. `CurrentUser/Elements/{elementId}/Links`

Returns links created by authors in the group.

Format:

`<ArrayOfLinks>
 <Link Name="<Name of link>" Description="<Description of link>" Url="<Url of link>" />
 ...
</ArrayOfLinks>`

You shouldn't assume that `Url` is parsable as a URI, as it has been entered by a user.

Example:

```
<ArrayOfLinks>
  <Link Name="XKCD" Description="" Url="http://xkcd.com/" />
  <Link Name="Ordbogen.com" Description="Ubegrænset antal opslag"
Url="http://www.ordbogen.com.globalproxy.cvt.dk/" />
</ArrayOfLinks>
```

4.25. `CurrentUser/Elements/{elementId}/Messages (POST)`

The format is likely to change at some point!

Creates a new message in the specified group. Post data (application/x-www-urlform-encoded):

- `subject`: The subject of the message. Max length is 100 characters.
- `content`: The content of the message. This is supposed to be an XHTML-fragment without html or body tags. It does not need to have a root element. It will be cleaned server-side so that script tags etc. are removed. The server will also try to create valid XHTML from the input.
- `sendSms`: Optional parameter specifying whether the subject of this message should be sent as an SMS to participants in this group. Only participants who have enabled SMS in this group will receive it – it works just like in the web edition of CN. The value must be “true” or “false”.

Note that not all groups allow sending SMS; at ASB no groups at all allow it. You will have to call `CurrentUser/Elements/{elementId}` first and check whether SMS is allowed.

Example:

```
POST /CurrentUser/Elements/207935/Messages HTTP/1.1
Host: www.campusnet.dtu.dk
Content-Type: application/x-www-form-urlencoded
Content-Length: 101
Authorization: Basic blabla
```

```
subject=Hello%2C+Dave.&content=I'm+sorry%2C+Dave.%3Cbr%2F%3EI'm+afraid+I+can't+do+that.&sendSms=false
```

Errors:

- 400, `ArgumentException` if `subject` or `content` is not specified. Also occurs when subject is too long. Also occurs if `sendSms` is specified but cannot be parsed.
- 403, `InsufficientRights` if you don't have author or administrator rights on the group.
- 403, `SmsNotAllowed` if `sendSms` was true and the current group does not allow sending SMS.
- 403, `DoublePost` if you try to send a message identical to one sent recently to the same group.

4.26. `CurrentUser/Elements/{elementId}/Messages/Newest`

Returns 10 newest messages from specified group.

Format: see 4.34.

4.27. `CurrentUser/Elements/{elementId}/Messages/Newest/From/{first}/Returning/{amount}`

Returns `amount` newest messages from specified group, starting from `first`.

Errors: see 4.47.

Format: see 4.34.

4.28. CurrentUser/Elements/{elementId}/Participants

Returns a list of participants (users) in a group.

Format:

```
<ArrayOfUserElementRelation>
  <UserElementRelation ACL="<ACL is described below>">
    <User GivenName="<Given name of the user. Middle names are included here.>"
    FamilyName="<Family name of the user.>" UserId="<Int32 id of the user>" Closed="<Is the user
closed? Possible values are true and false>" Email="<Email address of user>"
PreferredLanguage="<Preferred language in CampusNet. Can be da-DK or en-GB>" UserName="<The
users username>" StudentId="<The stads identification number>" />
  </UserElementRelation>
  ...
</ArrayOfUserElementRelation>
```

ACL: Access control list is a name identifying the rights a user has on the group. Possible values are Administrator, Author and User. Generally a user having ACL = User cannot edit anything. The two other levels can edit element data.

Example:

```
<ArrayOfUserElementRelation>
  <UserElementRelation ACL="Administrator">
    <User GivenName="Anders" FamilyName="And" UserId="1234" Closed="false"
Email="aa@adm.dtu.dk" PreferredLanguage="da-DK"/>
  </UserElementRelation>
  <UserElementRelation ACL="User">
    <User GivenName="Mickey" FamilyName="Mouse" UserId="5678" Closed="true"
Email="bounce@campusnet.dtu.dk" PreferredLanguage="en-GB" UserName="aand"
StudentId="s220102" />
  </UserElementRelation>
</ArrayOfUserElementRelation>
```

4.29. CurrentUser/Elements/Search?query={query}&first={first}&amount={amount}

Performs a search for your active groups in the CampusNet database using `query` as the search string. Mainly just the group name and front-page text is used for the search. The search algorithm supports double quotes for exact matches and the exclusion operator “-”.

The parameters `first` and `amount` control which portion of the search results are shown. These are optional and default to 1 and 20, respectively. The maximum value for both values is 100.

If more parameters are added to the specification at some point, they will be optional as well.

Format: Atom

Example URL: `CurrentUser/Elements/Search?query=math&first=1&amount=10`

Errors: See 4.3.

4.30. CurrentUser/Files

Returns all personal files for the current user. The format is the same as specified in 4.18 with the exception that `CreatorId` is not returned since it is irrelevant.

4.31. CurrentUser/Files/{fileId}/Bytes

Returns the file contents of a personal file with a specific `fileId`.

This resource accepts the URL query string parameters `serviceUsername` and `servicePassword`.

Format: see 4.19.

4.32. CurrentUser/Files/{fileId}/Versions/{version}/Bytes

Returns the file contents of a specific version of a file as indicated by `fileId` and `version`.

This resource accepts the URL query string parameters `serviceUsername` and `servicePassword`.

Format: see 4.19.

4.33. CurrentUser/Files/Newest

Deprecated – use `CurrentUser/Files/Newest/From/{first}/Returning/{amount}` instead.

Returns info about 10 newest files from integrated groups.

Format: see 3.5.

4.34. CurrentUser/Files/Newest/From/{first}/Returning/{amount}

Returns info about `amount` newest files from integrated groups, starting from `first`.

Format: see 3.5.

4.35. CurrentUser/Files/Search?query={query}&first={first}&amount={amount}

Performs a search for files in your active groups in the CampusNet database using `query` as the search string. File and group names are used for the search but not the actual file content. The search algorithm supports double quotes for exact matches and the exclusion operator “-”.

The parameters `first` and `amount` control which portion of the search results are shown. These are optional and default to 1 and 20, respectively. The maximum value for both values is 100.

If more parameters are added to the specification at some point, they will be optional as well.

Format: Atom

Example URL: `CurrentUser/Files/Search?query=report&first=1&amount=10`

Errors: See 4.3.

4.36. CurrentUser/Files/Upload?folderId={folderId} (POST)

Works as 4.22 but uploads to the personal file sharing.

4.37. CurrentUser/Grades

This currently only works for DTU:

Returns all grades for the user. Results can be credit (da.: merit) or regular results such as exams, projects and so on.

Format:

```
<EducationProgrammes>
  <EducationProgramme DisplayName="<Name of the education programme>" Active="<Boolean
specifying whether this education programme is currently active for this user>">
    <PassedEctsSum Exams="<Float ECTS points not acquired from credit>" Credits="<Float ECTS
points acquired from credit>" Total="<Float Total ECTS points acquired in this programme>" />
    <ExamResults>
      <ExamResult CourseCode="<String course code>" EctsGiven="<Boolean specifying whether
ECTS points for this course where given or not. Most times if a course was passed, points
were given and vice versa but it is not always the case>" EctsPoints="<Float specifying
number of ECTS points>" Grade="<String grade>" Period="<String 'Summer' or 'Winter'. More
values may be added here later on>" Year="<Integer year the course was taken>" Name="<Name
of the course>" />
      ...
    </ExamResults>
    <CreditResults>
      <CreditResult EctsGiven="<Boolean specifying whether ECTS points for this course where
given or not. Most times if a course was passed, points were given and vice versa but it is
not always the case>" EctsPoints="<Float specifying number of ECTS points>" Grade="<String
grade>" Date="<Date the credit was given>" Name="<Name of the credit>" />
      ...
    </CreditResults>
  </EducationProgramme>
  ...
</EducationProgrammes>
```

The grades in CampusNet are just a copy of the grades in the official system. If they get out of sync, the sum of ECTS-credits in CampusNet is not the same as the sum of credits in the official system. In this case the PassedEctsSum element is not returned.

Some names can be returned both in Danish and English. Use the `accept-language` header to specify either da-DK or en-GB. English is default.

Example:

```
<EducationProgrammes>
  <EducationProgramme DisplayName="Diplom (Dip02)" Active="true">
    <PassedEctsSum Exams="115" Credits="35" Total="150" />
    <ExamResults>
      <ExamResult CourseCode="41535" EctsGiven="true" EctsPoints="7.5" Grade="10"
Period="Summer" Year="2011" Name="Dynamics" />
      <ExamResult CourseCode="41540" EctsGiven="true" EctsPoints="10" Grade="10"
Period="Summer" Year="2011" Name="Mechanical Engineering Design" />
      <ExamResult CourseCode="41534" EctsGiven="false" EctsPoints="5" Grade="0"
Period="Winter" Year="2009" Name="Materials Science" />
    </ExamResults>
  </EducationProgramme>
</EducationProgrammes>
```

```

    </ExamResults>
    <CreditResults>
      <CreditResult EctsGiven="true" EctsPoints="5" Grade="BE" Date="2004-08-19" Name="41782
Produktionsteknologi, øvelser (erhvervserfaring)"/>
      <CreditResult EctsGiven="true" EctsPoints="30" Grade="BE" Date="2004-08-10"
Name="Ingeniørpraktik (Elektrikeruddannelsen i Danmark)"/>
    </CreditResults>
  </EducationProgramme>
  <EducationProgramme DisplayName="Without education" Active="false">
    <ExamResults>
      <ExamResult CourseCode="10020" EctsGiven="false" EctsPoints="10" Grade="-3"
Period="Winter" Year="2010" Name="Physics 1"/>
      <ExamResult CourseCode="10020" EctsGiven="false" EctsPoints="10" Grade="00"
Period="Summer" Year="2010" Name="Physics 1"/>
    </ExamResults>
    <CreditResults/>
  </EducationProgramme>
</EducationProgrammes>

```

4.38. CurrentUser/InformationSources

Returns metadata about the information sources that are available on the frontpage of CampusNet. Different universities have different sources and they may change over time.

Format:

```

<InformationSources>
  <InformationSource Id="<String id of the source>" Name="<Name of the source>"
PngIcon="<Base 64 encoded PNG icon representing the feed>"/>
  ...
</InformationSources>

```

Both Danish and English names can be returned. Use the `accept-language` header to specify either `da-DK` or `en-GB`. English is default.

The icon is guaranteed to always be square but we make no guarantees as to the size of the icon – when using it your application must resize as necessary. If the image doesn't use the whole space (e.g. rounded corners), transparency is used in the icon.

Example:

```

<InformationSources>
  <InformationSource Id="PortalMessages" Name="Portal messages" PngIcon="iVBOR...RK5CYII="/>
  <InformationSource Id="StudyMessages" Name="Study messages" PngIcon="iVBOR...TkSuQmCC"/>
  <InformationSource Id="NoticeBoard" Name="Notice board" PngIcon="iVBORw...rkJggg="/>
</InformationSources>

```

4.39. CurrentUser/InformationSources/{id}

Returns a news feed from the information source specified by `id`. The Atom format is used for this.

Errors:

- 404, `InformationSourceNotFound` if the source does not exist or the user doesn't have access to it.

- 503, `InformationSourceException` if the source currently does not work.

4.40. `CurrentUser/Library/Feed`

This currently only works for ASB:

Returns a news feed from the Library.

Format: Atom

4.41. `CurrentUser/Library/Holds`

This currently only works for ASB:

Returns information about material ready for pickup at the library.

Format:

```
<Holds>
  <Hold Author="<string name of the author>" HoldShelfEndDate="<ISO date specifying when the
book will no longer be on hold>" MaterialType="<string material type>"
PickupLocation="<string denoting where to pickup the material>" Title="<string title of the
material>" PickupNumber"<optional string denoting the pickup number>" Year="<optional int
specifying a year for the material>" />
  ...
</Holds>
```

Example:

```
<Holds>
  <Hold Author="Lassen, Linda, f. 1948" HoldShelfEndDate="2011-10-12T00:00:00"
MaterialType="Books" PickupLocation="Statsbiblioteket" Title="Myter og facts om overvægt"
PickupNumber="3-77" Year="2008" />
</Holds>
```

Errors:

- 404, `NoCprNumber` if CampusNet does not know the cpr number of the user.
- 404, `NotALibraryUser` if the library does not recognize the user.

In both cases error XML is with more information about how to enable library integration. This information is localized according to the `accept-language` header and presented in the description element of the error xml. It is some kind of HTML/XHTML and does not necessarily validate. For example, the error could look like this:

```
<Error xmlns="http://schemas.datacontract.org/2004/07/Arcanic.CampusNet.DataService">
  <Description>
    <lt;p>&#xD;
    You do not have access to the <lt;b>My Library</b> service.</p>&#xD;
    <lt;ol>&#xD;
    <lt;li>&#xD;
    You are not registered as a library user with a Danish CPR number. This service only
works with a valid Danish CPR number.&#xA0;Please contact the library if you have any
questions. Call +45 89 48 65 20</li>&#xD;
    <lt;/ol>&#xD;
```



```
</Description>
<Name>NotALibraryUser</Name>
</Error>
```

4.42. CurrentUser/Library/Information

This currently only works for ASB:

Returns general information about the library such as opening hours, addresses, contact info and so on.

Format:

HTML (not guaranteed to be valid and it is unknown whether it is HTML 4, 5, or XHTML.)

Both Danish and English text can be returned. Use the `accept-language` header to specify either da-DK or en-GB. English is default.

Example:

```
<p>
  <a href="http://www.asb.dk/somepage.aspx">ASB Library opening hours</a>
</p>
<p>
  If you are not a user of the library, register at the website.
</p>
```

4.43. CurrentUser/Library/Loans

This currently only works for ASB:

Returns information about currently active loans for the user.

Format:

```
<Loans>
  <Loan Author="<string name of the author>" DueDate="<ISO date. The book must be handed in
before this date>" LoanDate="<ISO date that the book was borrowed>" MaterialType="<string
material type>" Title="<string title of the material>" Year="<optional int specifying a year
for the material>" />
  ...
</Loans>
```

Example:

```
<Loans>
  <Loan Author="Nord, Christiane" DueDate="2011-11-07T00:00:00" LoanDate="2011-09-
12T00:00:00" MaterialType="Books" Title="Text analysis in translation"/>
  <Loan Author="Griffiths, Patrick, 1942-" DueDate="2011-11-07T00:00:00" LoanDate="2011-09-
12T00:00:00" MaterialType="Books" Title="An introduction to English semantics and
pragmatics" Year="2008"/>
</Loans>
```

Errors: see 4.41.

4.44. CurrentUser/Library/Requests

This currently only works for ASB:

Returns information about currently active reservations for the user.

Format:

```
<Requests>
  <Request Author="<string name of the author>" LastInterestDate="<ISO date. After this date
the user is automatically removed from the queue>" MaterialType="<string material type>"
PickupLocation PickupLocation="<string denoting where to pickup the material>"
QueuePosition="<int position in the queue>" Title="<string title of the material>"
Year="<optional int specifying a year for the material>" />
  ...
</Requests>
```

Example:

```
<Requests>
  <Request Author="Lassen, Linda, f. 1948" LastInterestDate="2011-11-26T00:00:00"
MaterialType="Books" PickupLocation="Statsbiblioteket" QueuePosition="1"
Title="Overspisning" />
  <Request Author="Cornelissen, Joep" LastInterestDate="2011-12-07T00:00:00"
MaterialType="Books" PickupLocation="Statsbiblioteket" QueuePosition="5" Title="Corporate
communication" Year="2011" />
</Requests>
```

Errors: see 4.41.

4.45. CurrentUser/Links

Returns the links that the user has created in CampusNet.

Format:

```
<ArrayOfLinks>
  <Link Name="<Name of link>" Description="<Description of link>" Url="<Url of link>" />
  ...
</ArrayOfLinks>
```

You shouldn't assume that `Url` is parsable as a URI, as it is entered by the user.

Example:

```
<ArrayOfLinks>
  <Link Name="XKCD" Description="" Url="http://xkcd.com/" />
  <Link Name="Ordbogen.com" Description="Ubegrænset antal opslag"
Url="http://www.ordbogen.com.globalproxy.cvt.dk/" />
</ArrayOfLinks>
```

4.46. CurrentUser/Messages/Newest

Returns 10 newest messages from integrated groups.

Format: By giving an `accept` header when requesting, you can choose between formats. The default format is text/html.

4.46.1. accept: text/html

This will give you the following format

```
<Messages>
  <GroupMessage Id="<int32 unique id of the message>" ElementId="<int32 id of the group to
which the message belongs>" AuthorUserId="<int32 id of the user who last edited the
message>" Updated="<ISO date designating when the message was created/last changed>"
Subject="<Subject of the message>" Body="<Body of the message in HTML format>" />
  ...
</Messages>
```

Example:

```
<Messages>
  <GroupMessage Id="123" ElementId="456" AuthorUserId="789" Updated="2010-11-
01T15:11:30.257" Subject="Mødet mandag den 8. november"
Body="&lt;p&gt;Hej&lt;/p&gt;&lt;p&gt;&nbsp;&lt;/p&gt;&lt;p&gt;S&aring; er der
m&oslash;de&lt;/p&gt;" />
</Messages>
```

4.46.2. accept: text/plain

This gives you the same format as text/html but with Messages/GroupMessage/@Body containing plain text instead of HTML.

Example:

```
<Messages>
  <GroupMessage Id="123" ElementId="456" AuthorUserId="789" Updated="2010-11-
01T15:11:30.257" Subject="Mødet mandag den 8. november" Body="Hej&#xA;&#xA;Så er der møde" />
</Messages>
```

4.46.3. accept: application/atom+xml

This gives you data in the Atom format. Body content in this case is HTML.

4.47. CurrentUser/Messages/Newest/From/{first}/Returning/{amount}

Returns `amount` newest messages integrated groups, starting from `first`.

Errors:

- 400, `FormatException` if `first` or `amount` could not be parsed as integers.
- 400, `ArgumentException` if `first` or `amount` are not positive.

Format: see 4.34.

4.48. CurrentUser/Messages/Search?query={query}&first={first}&amount={amount}

Performs a search for messages that are not expired. It searches in your active groups in the CampusNet database using `query` as the search string. The search algorithm supports double quotes for exact matches and the exclusion operator “-”.

The parameters `first` and `amount` control which portion of the search results are shown. These are optional and default to 1 and 20, respectively. The maximum value for both values is 100.

If more parameters are added to the specification at some point, they will be optional as well.

Format: Atom

Example URL: `CurrentUser/Messages/Search?query=example&first=1&amount=10`

Errors: See 4.3.

4.49. CurrentUser/UserInfo

Returns information about the current user.

Format: By giving an `accept` header when requesting, you can choose between formats. The default format is `application/xml`.

4.49.1. accept: application/xml

This will always return information about the current user.

Format:

```
<User GivenName="<Given name of the user. Middle names are included here.>"
FamilyName="<Family name of the user.>" UserId="<Int32 id of the user>" Closed="<Is the user
closed? Possible values are true and false>" Email="<Email address of user>"
PreferredLanguage="<Preferred language in CampusNet. Can be da-DK or en-GB>"/>
```

Example:

```
<User GivenName="Anders" FamilyName="And" UserId="1234" Closed="false" Email="aa@adm.dtu.dk"
PreferredLanguage="da-DK"/>
```

4.49.2. accept: text/x-vcard

This currently only works for DTU:

This returns data in the vCard 3.0 format. Some fields such as `TITLE` are translated according to the language specified in the `accept-language` header (da-DK or en-GB). English is default. Not all users have a vCard – an error is returned if that is the case. Note that the country name can take any value as it does not come from a closed list.

Example:

```
BEGIN:VCARD
VERSION:3.0
N:Ørding-Thomsen;Martin
FN:Martin Ørding-Thomsen
ORG:Arcanic A/S
TITLE:Udvikler
PHOTO;VALUE=URL;TYPE=GIF:http://www.dtubasen.dtu.dk/showimage.aspx?id=22851
TEL;TYPE=CELL:
TEL;TYPE=WORK:45257450
TEL;TYPE=WORK:
ADR;TYPE=WORK:;;;;dk
LABEL;TYPE=WORK:\n \ndk
EMAIL;TYPE=PREF,INTERNET:mat@arcanic.dk
REV:20110414161100
END:VCARD
```

Errors:

- 406, `NoVCard` if `accept: text/x-vcard` was specified and the user has no vCard.

4.50. `CurrentUser/Users/{userId}`

Returns information about the user specified by `userId`.

Format: see 4.48.

Errors:

- 400, `BadUserIdFormat` if `userId` could not be parsed.
- 404, `UserNotFound` if the user doesn't exist.
- 406, `NoVCard` if `accept: text/x-vcard` was specified and the user has no vCard.

4.51. `CurrentUser/Users/{userId}/Picture`

Returns a picture of the user identified by `userId`. The picture has the maximum size allowed by the system. If a user has no picture, a blank image is returned.

Format: JPEG or GIF.

The headers include `content-length`, and type of picture is specified in the `content-type` header.

Errors:

- 400, `BadUserIdFormat` if `userId` could not be parsed.
- 404, `UserNotFound` if the user doesn't exist.

4.52. `CurrentUser/Users/{userId}/Picture/Width/{width}`

Returns a picture of the user identified by `userId`. The picture is stretched according to the `width` parameter. If a user has no picture, a blank image is returned.

Format: JPEG or GIF.

The headers include `content-length`. The picture type is specified in the `content-type` header.

Errors:

Same as in 4.51 plus the following:

- 400, `BadUserIdFormat` if `userId` could not be parsed.
- 400, `BadWidthFormat` if `width` could not be parsed.

4.53. `CurrentUser/Users/Search?query={query}&first={first}&amount={amount}`

Performs a search among active users in the CampusNet database using `query` as the search string. The only fields included in the search are first name, last name, username, and email. At some point more fields will most likely be included in the search. Additionally, search behavior and ranking is likely to change

without warning. The search algorithm supports double quotes for exact matches and the exclusion operator “-”.

Each university has its own set of rules governing which users are included in the search. These rules are likely to change at any time.

The parameters `first` and `amount` control which portion of the search results are shown. These are optional and default to 1 and 20, respectively. The maximum value for both values is 200.

If more parameters are added to the specification at some point, they will be optional as well.

Format:

```
<ArrayOfUsers>
  <User GivenName="<Given name of the user. Middle names are included here.>"
  FamilyName="<Family name of the user.>" UserId="<Int32 id of the user>" Closed="<Is the user
closed? Possible values are true and false>" Email="<Email address of user>"
PreferredLanguage="<Preferred language in CampusNet. Can be da-DK or en-GB>" />
</ArrayOfUsers>
```

It is likely that more information will be included in the search result at some point, most likely both additional elements and attributes. You should take this into account when constructing your parser.

Example:

Performing query: `CurrentUser/Users/Search?query=maximilian&first=1&amount=10`

```
<ArrayOfUsers>
  <User GivenName="Christian Maximilian" FamilyName="Jensen" UserId="103842" Closed="false"
Email="s123@student.dtu.dk" PreferredLanguage="en-GB" />
  <User GivenName="Maximilian" FamilyName="Hansen" UserId="142332" Closed="false"
Email="s456@student.dtu.dk" PreferredLanguage="da-DK" />
  <User GivenName="Maximilian" FamilyName="Josef von Steiner" UserId="128028" Closed="false"
Email="s789@student.dtu.dk" PreferredLanguage="en-GB" />
</ArrayOfUsers>
```

Errors:

- 400, `NoQuery` if `query` is empty or not specified at all.
- 400, `FormatException` if `first` or `amount` is specified but could not be parsed as an integer.
- 400, `ArgumentException` if `first` or `amount` is specified but is not positive.
- 400, `ArgumentException` if `first` or `amount` is specified but is higher than the maximum allowed value.
- 400, `BadQuery` if the search could not be completed. This is normally due to all search terms being ignored for one reason or another.
- 504, `Timeout` if performing the search took too long. This doesn't seem to happen currently, but may happen later if the search algorithm is changed and searches for very short string are performed.

4.54. CurrentUser/Polls/From/{first}/To/{amount}

Returns polls that the user can access. Skips {first} inactive polls, returns {amount} inactive polls and all active polls. The GET-parameter excludeActive can be used to exclude the active polls (default value is false).

Errors:

CODE	NAME	REASON
400	BadFirstFormat	Parameter first must be an integer
400	BadAmountFormat	Parameter amount must be an integer

Format:

```
<Polls>
  <ActivePolls>
    <Poll Id="int" Question="string" ElementId="int" AuthorUserId="int"
Created="datetime" IsActive="bool" ResultsVisibility="Visible|VisibleAfterVoting|NotVisible"
Closing="datetime" CurrentUserHasVoted="bool" CurrentUserCanVote="bool"
Type="SingleVote|MultipleVotes">
      <Element Id="int" ParentId="int" Name="string" ContextName="string"
IsArchived="bool" SubgroupCount="int"/>
      <User GivenName="string" FamilyName="string" UserId="int" Closed="bool"
Email="string" PreferredLanguage="string"/>
    </Poll>
    ...
  </ActivePolls>
  <InactivePolls>
    <!-- Same as above -->
  </InactivePolls>
</Polls>
```

Example response:

```
<Polls>
  <ActivePolls>
    <Poll Id="41" Question="ytytytyt" ElementId="418005" AuthorUserId="1234"
Created="2014-05-30T10:49:01.127" IsActive="true" ResultsVisibility="VisibleAfterVoting"
Closing="2014-07-09T17:49:50.863" CurrentUserHasVoted="true" CurrentUserCanVote="false"
Type="MultipleVotes">
      <Element Id="418005" ParentId="3063" Name="Android app testgruppe"
ContextName="DTU Bibliotek/LRClassic/CN-følgegruppe/Android app testgruppe"
IsArchived="false" SubgroupCount="0"/>
      <User GivenName="Fornavn" FamilyName="Efternavn" UserId="1234" Closed="false"
Email="email@email.dk" PreferredLanguage="da-DK"/>
    </Poll>
  </ActivePolls>
  <InactivePolls>
    <Poll Id="40" Question="Test" ElementId="418005" AuthorUserId="1234" Created="2014-
05-30T10:48:37.533" IsActive="false" ResultsVisibility="NotVisible" Closing="2014-06-
05T12:13:26.67" CurrentUserHasVoted="false" CurrentUserCanVote="false" Type="SingleVote">
      <Element Id="418005" ParentId="3063" Name="Android app testgruppe"
ContextName="DTU Bibliotek/LRClassic/CN-følgegruppe/Android app testgruppe"
IsArchived="false" SubgroupCount="0"/>
      <User GivenName="Fornavn" FamilyName="Efternavn" UserId="1234" Closed="false"
Email="email@email.dk" PreferredLanguage="da-DK"/>
    </Poll>
  </InactivePolls>
</Polls>
```

```

    </Poll>
  </InactivePolls>
</Polls>

```

4.55. CurrentUser/Polls/{id}

Returns information about the poll with the id {id} and its options. If the user has cast a vote, the CurrentUserHasVotedOnThis attribute is present on the Option elements.

Errors:

CODE	NAME	REASON
400	BadIdFormat	Parameter id must be an integer
403	NoAccess	User does not have access to the poll
404	NotFound	Poll not found

Format:

```

<Poll Id="int" Question="string" ElementId="int" AuthorUserId="int" Created="datetime"
IsActive="bool" ResultsVisibility="Visible|VisibleAfterVoting|NotVisible" Closing="datetime"
CurrentUserHasVoted="bool" CurrentUserCanVote="bool" Type="SingleVote|MultipleVotes">
  <Element Id="int" ParentId="int" Name="string" ContextName="string" IsArchived="bool"
SubgroupCount="int" />
  <User GivenName="string" FamilyName="string" UserId="int" Closed="bool" Email="string"
PreferredLanguage="string" />
  <Options>
    <Option Id="int" Text="string" CurrentUserVotedOnThis="bool" />
    ...
  </Options>
</Poll>

```

Example response:

```

<Poll Id="41" Question="ytytytyt" ElementId="418005" AuthorUserId="1234" Created="2014-05-
30T10:49:01.127" IsActive="true" ResultsVisibility="VisibleAfterVoting" Closing="2014-07-
09T17:49:50.863" CurrentUserHasVoted="true" CurrentUserCanVote="false" Type="MultipleVotes">
  <Element Id="418005" ParentId="3063" Name="Android app testgruppe" ContextName="DTU
Bibliotek/LRClassic/CN-følgegruppe/Android app testgruppe" IsArchived="false"
SubgroupCount="0" />
  <User GivenName="Fornavn" FamilyName="Efternavn" UserId="1234" Closed="false"
Email="email@email.dk" PreferredLanguage="da-DK" />
  <Options>
    <Option Id="106" Text="rtyrtyrtyrty" CurrentUserVotedOnThis="true" />
    <Option Id="107" Text="werwerwerwe" CurrentUserVotedOnThis="true" />
    <Option Id="108" Text="wqqweqweqwe" CurrentUserVotedOnThis="false" />
  </Options>
</Poll>

```

4.56. CurrentUser/Polls/{id}/Results

Returns information about the poll with the id {id} and its options, including the vote count and correct answers. If the user has cast a vote, the CurrentUserHasVotedOnThis attribute is present on the Option elements.

Errors:

CODE	NAME	REASON
400	BadIdFormat	Parameter id must be an integer
403	NoAccess	User does not have access to the poll
404	NotFound	Poll not found
403	NotVisible	Poll results are not visible

Format:

```
<Poll Id="int" Question="string" ElementId="int" AuthorUserId="int" Created="datetime"
IsActive="bool" ResultsVisibility="Visible|VisibleAfterVoting|NotVisible" Closing="datetime"
CurrentUserHasVoted="bool" CurrentUserCanVote="bool" Type="SingleVote|MultipleVotes">
  <Element Id="int" ParentId="int" Name="string" ContextName="string" IsArchived="bool"
SubgroupCount="int"/>
  <User GivenName="string" FamilyName="string" UserId="int" Closed="bool" Email="string"
PreferredLanguage="string"/>
  <Options>
    <Option Id="int" Text="string" Votes="int" IsCorrectAnswer="bool"
CurrentUserVotedOnThis="bool"/>
    ...
  </Options>
</Poll>
```

Example response:

```
<Poll Id="41" Question="ytytytyt" ElementId="418005" AuthorUserId="1234" Created="2014-05-
30T10:49:01.127" IsActive="true" ResultsVisibility="VisibleAfterVoting" Closing="2014-07-
09T17:49:50.863" CurrentUserHasVoted="true" CurrentUserCanVote="false" Type="MultipleVotes">
  <Element Id="418005" ParentId="3063" Name="Android app testgruppe" ContextName="DTU
Bibliotek/LRClassic/CN-følgegruppe/Android app testgruppe" IsArchived="false"
SubgroupCount="0"/>
  <User GivenName="Fornavn" FamilyName="Efternavn" UserId="1234" Closed="false"
Email="email@email.dk" PreferredLanguage="da-DK"/>
  <Options>
    <Option Id="106" Text="rtyrtyrtyrty" Votes="2" IsCorrectAnswer="true"
CurrentUserVotedOnThis="true"/>
    <Option Id="107" Text="werwerwerwe" Votes="1" IsCorrectAnswer="true"
CurrentUserVotedOnThis="true"/>
    <Option Id="108" Text="wqqweqweqwe" Votes="1" IsCorrectAnswer="false"
CurrentUserVotedOnThis="false"/>
  </Options>
</Poll>
```

4.57. CurrentUser/Polls/{id}/Vote (POST)

Places a vote on the option ids given in the request body (comma-separated in the optionIds key) for the poll with id {id}. If the poll only allows single-votes the optionIds parameter may only contain one option ID.

Errors:

CODE	NAME	REASON
400	BadOptionsIdFormat	Parameter optionIds must be a comma-separated list of integers
400	BadIdFormat	Parameter id must be an integer
404	NotFound	Poll not found
403	NoAccess	User does not have access to the poll

403	NoVoteAccess	User is not able to vote on the poll
400	TooManyVotes	Poll does not allow multiple votes

Format:

HTTP 200 with no content if successful.

Example request body:

optionIds=1,3

4.58. CurrentUser/Events

Gets all events that the current user is associated with either by being an administrator on the event or by participating in the event as a user.

Format:

```
<Events>
  <Event Title="string" Participants="int" CnElement="int" CnElementUrl="string"
ACL="Administrator|User" SignupDeadline="datetime" MaxParticipants="int">
    <Description>"string"</Description>
    <EventDates>
      <EventDate Title="string" Place="string" StartTime="datetime" EndTime="datetime"/>
    </EventDates>
  </Event>
</Events>
```

An Event element always has a Title, Participants, CnElement, CnElementUrl and ACL attribute. SignupDeadline and MaxParticipants are optional and may not occur.

An Event will always contain a Description child element and one or more EventDates.

An EventDate will always contain a Title, Place, StartTime and EndTime attributes.

Additionally an event will return the defined recommendation targets used to expose the event to potential users. This will only be included if the current user is an administrator for the event.

...

```
<RecommendationTargets>
  <User UserName="string" GivenName="string" FamilyName="string"/>
  <Group CnElementId="int" CnElementName="string"/>
  <TargetAudience Employees="bool" Students="bool">
    <AudienceGroup Id="int"/>
  </TargetAudience>
</RecommendationTargets>
```

...

The RecommendationTargets element can contain one or more User, Group or TargetAudience elements with attributes as shown above. Each TargetAudience can contain one or more AudienceGroup elements.

4.59. CurrentUser/Events/Recommended

Returns all recommended events for the CurrentUser.

Format: same as [CurrentUser/Events](#)

4.60. CurrentUser/Events/Archived

Returns all archived events where the CurrentUser is associated with either as a user or administrator.

Format: same as [CurrentUser/Events](#)

5. Beta functions.

The following functions are not complete and meant for testing only. They will most likely change in the near future.

The documentation currently lack error codes and their meaning.

All functions that take an input will accept input as either `application/json` or `application/xml`. The preferred method is set in the header field `Content-type`.

Likewise, the return format for functions that return content can be either `application/json` or `application/xml`. This is set using the `Accept: header field`.

5.1. CurrentUser/Elements/{elementId}/Messages/v2 (GET)

Returns an array of messages from the specified group. The format of the output can be defined by setting the accept header field.

Accept: `application/json`

```
{
  "Id": 51620,
  "Title": "api update2",
  "CreationTime": "2015-06-09T16:46:23.223",
  "LastView": "2015-06-10T15:49:25.59",
  "Content": {
    "ContentType": "Html",
    "Value": "content update2"
  },
  "Author": {
    ""
  },
  "Element": {
    "Id": 309,
    "Name": "Planning for Math"
  },
  "LastEdit": {
    "Comment": "tjooooo jeg skulle lige teste",
    "Time": "2015-06-10T15:49:25.457",
    "Person": {
      ""
    }
  },
  "Comments": [
```

```

    ...
  },
  "LastAction": {
    "Time": "2015-06-10T15:49:25.457",
    "Type": "MessageCreated",
    "Person": {
      ...
    }
  },
  "MessageType": "Sticky",
  "Permission": {
    "CanCreateComment": true
  },
  "ValidUntill": "2016-04-04T00:00:00",
  "HighPriority": false
}

```

The person, author and comments section is removed for brevity.

Accept: application/xml

```

<?xml version="1.0" encoding="utf-16"?>
  <ArrayOfMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <Message Id="51620" Title="api update2" CreationTime="2015-06-09T16:46:23.223"
      MessageType="Sticky">
      <LastView>2015-06-10T15:49:25.59</LastView>
      <Content ContentType="Html">content update2</Content>
      <Author FirstName="Lars" LastName="Kampp Zilmer" Id="6" />
      <Element Id="309" Name="Planning for Math" />
      <LastEdit Time="2015-06-10T15:49:25.457">
        <Person FirstName="Lars" LastName="Kampp Zilmer" Id="6" />
        <Comment>tjooooo jeg skulle lige teste</Comment>
      </LastEdit>
      <Comments>
        <Comment Id="19753" CreationTime="2015-06-10T14:37:22.02">
          <LastEdit Time="2015-06-10T14:37:36.537">
            <Person FirstName="Lars" LastName="Kampp Zilmer" Id="6" />
          </LastEdit>
          <Author FirstName="Lars" LastName="Kampp Zilmer" Id="6" />
          <Content ContentType="Html">kommetar :) nu med ændring </Content>
        </Comment>
      </Comments>
      <LastAction Time="2015-06-10T15:49:25.457" Type="MessageCreated">
        <Person FirstName="Lars" LastName="Kampp Zilmer" Id="6" />
      </LastAction>
      <Permission CanCreateComment="true" />
      <ValidUntill>2016-04-04T00:00:00</Validuntill>
      <HighPriority>>false</HighPriority>
    </Message>
    ...
  </ArrayOfMessage>

```

5.2. [CurrentUser/Elements/{elementId}/Messages/v2/From/{first}/returning/{amount}/ \(GET\)](#)

Returns the set amount of messages from an element specified by the {first} and the {amount}. Format and accept-header options are the same as in [CurrentUser/Elements/{elementId}/Messages/v2 \(GET\)](#)

5.3. [CurrentUser/Elements/{elementId}/Messages/v2/{messageId} \(GET\)](#)

Returns the message with the specified messageId, format and accept-header options are the same as in [CurrentUser/Elements/{elementId}/Messages/v2 \(GET\)](#).

5.4. CurrentUser/Elements/{elementId}/Messages/v2 (POST)

Creates a new message in the specified group.

Post-data:

- `MessageType`: the type of message. (`Standard|Sticky`) if set to sticky the posting user has to be administrator of the group otherwise an error will be returned. Defaults to standard.
- `Title`: The title/subject for the message.
- `Content`: The content of the message.
- `ContentType`: The Type of content (`PlainText|Html`).

If the message type is `sticky` there some additional options that can be set.

- `HighPriority`: Boolean, defaults to false.
- `CanCreateComment`: Boolean, should the message be open for comments, defaults to yes.
- `ValidUntill`: Date ISO8601 , the message is valid until this date. The message has no expiry date if it is not defined.

The method accepts post data in two different formats; the post format is set in the Content-Type header and can be one of the following:

Content-Type: `application/json`

The formatting structure for a json post is a little different the url-encoding version. The `Content` object contains a value, which is the actual content and the `ContentType` for the `Value`. The `Permission` object contains the `CanCreateComment` property that works as described above.

Example:

```
{
  "Title": "this amazing thing happened",
  "Content": {
    "Value": "so i was out jogging...",
    "ContentType": "Html"
  },
  "MessageType": "Sticky",
  "HighPriority": true,
  "ValidUntill": "2016-01-01",
  "Permission": {
    "CanCreateComment": true
  }
}
```

Content-Type: `application/xml`

Like for JSON the xml content formatting is slightly different.

Example:

```
<Message Title="this amazing thing happened" MessageType="Sticky">
  <Content ContentType="Html">so I was out jogging...</Content>
  <Permission CanCreateComment="true" />
  <ValidUntil>2016-01-01T00:00:00</ValidUntil>
  <HighPriority>true</HighPriority>
</Message>
```

5.5. [CurrentUser/Elements/{elementId}/Messages/v2/{messageId}](#) (PUT)

Updates a message with new data, input format is the same as

[CurrentUser/Elements/{elementId}/Messages/v2 \(POST\)](#) with the exception that there is a possibility to add a comment to the update.

Additional post data:

- `EditComment`: comment for the edit.

Examples

Content-Type: application/json

The post object is extended with a `LastEdit` object with a `comment` property that contains the comment string.

Content-Type: application/xml

The Message entity contains a `LastEdit` element with an inner `comment` element that contains the comment string.

5.6. [CurrentUser/Elements/{elementId}/Messages/v2/{messageId}](#) (DELETE)

Deletes the message specified from the element.

5.7. [CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments \(GET\)](#)

Returns an array of comments associated with a specific message. As with the get messages function there is a choice of different output formats.

Accept: application/json

```
[
  {
    "Id": 19753,
    "CreationTime": "2015-06-10T14:37:22.02",
    "LastEdit": {
      "Time": "2015-06-10T14:37:36.537",
      "Person": {
        "FirstName": "Lars",
        "LastName": "kampp Zilmer",
        "Id": 6
      }
    },
    "Author": {
      "FirstName": "Lars",
      "LastName": "kampp Zilmer",
      "Id": 6
    },
  },
]
```

```

    "Content": {
      "ContentType": "Html",
      "Value": "kommetar :) nu med ændring "
    }
  }
]

```

Accept: application/xml

```

<?xml version="1.0" encoding="utf-16"?>
  <ArrayOfComment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <Comment Id="19753" CreationTime="2015-06-10T14:37:22.02">
      <LastEdit Time="2015-06-10T14:37:36.537">
        <Person FirstName="Lars" LastName="kampp Zilmer" Id="6" />
      </LastEdit>
      <Author FirstName="Lars" LastName="kampp Zilmer" Id="6" />
      <Content ContentType="Html">kommetar :) nu med ændring </Content>
    </Comment>
  </ArrayOfComment>

```

5.8. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments/{commentId} (DELETE)

Deletes a comment associated with a specific message Id.

5.9. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments (POST)

Post-data:

- **Content:** The content of the message.
- **ContentType:** The Type of content (PlainText|Html).

The method accepts post data in three different formats; the post format is set in the Content-Type header and can be one of the following:

Content-Type: application/json

The formatting structure for a json post is a little different the url-encoding version. The `Content` object contains a `value`, which is the actual content and the `ContentType` for the `Value`. The `Permission` object contains the `CanCreateComment` property that works as described above.

Example:

```

{
  "Content": {
    "Value": "so i was out jogging...",
    "ContentType": "Html"
  }
}

```

Content-Type: application/xml

Like for JSON the xml content formatting is slightly different.

Example:

```
<Content ContentType="Html">so I was out jogging...</Content>
```

5.10. [CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments/{commentId}](#)(PUT)

Updates a comment. Accepts the same input forms as

[CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/Comments \(POST\)](#)

5.11. [CurrentUser/Elements/{elementId}/Messages/v2/NotificationSettings](#)(GET)

Messages may have notification settings associated with them, the settings are set for the element and applies to all the messages associated with that specific element.

Currently there is a setting for notifications through email or sms, where the email service can send notifications when new messages are created or when new comments in the current messages are added.

The SMS service only supports sending text messages, creating a new message with the sticky/high priority option to true will trigger the sending of Sms messages to the users with the option set.

As with the get messages function there is a choice of different output formats.

Accept: application/json

```
{
  "ElementId": 309,
  "NotificationPreferences": [
    {
      "NotificationType": "Email",
      "NotifyOnSticky": true,
      "NotifyEnabled": true,
      "NotifyOnStrandard": false,
      "NotifyOnStickyComment": true,
      "NotifyOnStandardComment": false
    },
    {
      "NotificationType": "Sms",
      "NotifyOnSticky": true,
      "NotifyEnabled": true
    }
  ]
}
```

Accept: application/xml

```
<?xml version="1.0" encoding="utf-16"?>
  <NotificationSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" ElementId="309">
```



```

    <Preferences>
      <NotificationPreference xsi:type="EmailNotificationPreference" NotificationType="Email"
NotifyOnSticky="true" NotifyEnabled="true" NotifyOnStandard="false"
NotifyOnStickyComment="true" NotifyOnStandardComment="false" />
      <NotificationPreference xsi:type="SmsNotificationPreference" NotificationType="Sms"
NotifyOnSticky="true" NotifyEnabled="true" />
    </Preferences>
  </NotificationSettings>

```

5.12. CurrentUser/Elements/{elementId}/Messages/v2/NotificationSettings (PUT)

Allows users to update their notification settings.

Post-data:

- EmailNotifySticky: Notify by email if a new sticky message is created.
- EmailNotfiyEnabled: Enable or disable the notification rule.
- EmailNotifyStandard: Notify by email if a new message is created.
- EmailNotifyStandardComment: Notify by email when a new comment is posted.
- EmailNotifyStickyComment: Notify by email when a comment is made on a sticky message.
- SmsNotifyEnabled: enable or disable Sms notification.
- SmsNotifySticky: notify by Sms if a new sticky message is posted to the group.

Content-Type: application/json

```

{
  "NotificationPreferences": [
    {
      "NotificationType": "Email",
      "NotifyOnSticky": true,
      "NotifyOnStandard": true,
      "NotifyOnStandardComment": false,
      "NotifyOnStickyComment": false
    },
    {
      "NotificationType": "Sms",
      "NotifyOnSticky": true
    }
  ]
}

```

Content-Type: application/xml

```

<NotificationSettings xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
  <Preferences>
    <NotificationPreference xsi:type='EmailNotificationPreference' NotificationType='Email'
NotifyOnSticky='true' NotifyEnabled='true' NotifyOnStandard='true'
NotifyOnStickyComment='false' NotifyOnStandardComment='false' />
    <NotificationPreference xsi:type='SmsNotificationPreference' NotificationType='Sms'
NotifyOnSticky='true' NotifyEnabled='true' />
  </Preferences>
</NotificationSettings>

```

5.13. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/NotificationSettings (GET)

It is possible to configure a notification setting for a single message such that an email is sent if new comments are added. By default, the setting will inherit from the general notification setting for the element.

Accept: application/json

```
{
  "MessageId": 51621,
  "NotificationPreferences": [
    {
      "NotificationType": "Email",
      "NotifyOnStandardComment": true,
      "IsSpecific": true
    }
  ]
}
```

Accept: application/xml

```
<?xml version="1.0" encoding="utf-16"?>
  <MessageNotificationSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" MessageId="51621">
    <Preferences>
      <NotificationPreference xsi:type="MessageNotificationPreference"
        NotificationType="Email" NotifyOnStandardComment="true" IsSpecific="true" />
    </Preferences>
  </MessageNotificationSettings>
```

The `IsSpecific` attribute/property defines if the preference is inherited (false) from the general notification preference or if it is defined specifically for the message (true).

5.14. CurrentUser/Elements/{elementId}/Messages/v2/{messageId}/NotificationSettings (PUT)

Updates the notification setting for a specific message.

Post-data:

Content-Type: application/json

```
{
  "NotificationPreferences": [
    {
      "NotificationType": "Message",
      "NotifyOnStandardComment": true
    }
  ]
}
```

Content-Type: application/xml

```
<NotificationSettings xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
```

```
<Preferences>
  <NotificationPreference xsi:type='MessageNotificationPreference'
NotificationType='Email' NotifyEnabled='true' NotifyOnStandardComment='true' />
</Preferences>
</NotificationSettings>
```

5.15. **CurrentUser/Messages/v2/{messageId }/RegisterView (PUT)**

Registers a view of the message with messageId from the current user.

Post-data: none

Response:

HTTP 200 with no content if successful.