API Documentation Mobile City Memory

These web services can be reached via the URL “/services/” + the respective web service name. The parameters have to be added in the given order, separated by forward slashes.

Example: <http://localhost/services/locations/52.12456/14.24363>

**Bold parameters** are required, *cursive parameters* optional.

These services return data in JSON.

For PUT-Actions, the parameters should be sent in JSON-Format, except where noted differently.

# locations/

## GET

This lists all the locations saved in the database.

### URL-Parameters

None

### Return data

List of locations:

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |

## POST

Authentication Required

This creates a new location in the database.

### Parameters

|  |  |
| --- | --- |
| **latitude** | Latitude of the location |
| **longitude** | Longitude of the location |
| **title** | Title of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |

### Return data

Depending on whether the authentication succeeded or failed, either a 201 Created or a 403 Forbidden status code. If authentication succeeds, the following header data will be returned.

|  |  |  |
| --- | --- | --- |
| Location | locations/location**\_id** | URI of the newly created location |

# locations/story\_count/

## GET

This lists all the locations saved in the database. It also includes a list of IDs of saved stories to these locations.

### URL-Parameters

None

### Return data

List of locations:

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of IDs of attached stories |

# locations/stories/title

## GET

This lists all the locations saved in the database. It also includes a list of IDs of saved stories to these locations, their title, abstract and their location.

### URL-Parameters

None

### Return data

List of locations:

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | abstract | Abstract of the story | | location | ID of the location | |

# locations/**lat**/*maxlat*/**lon**/*maxlon*/

## GET

This retrieves all locations in a certain area.

### URL-Parameters

|  |  |
| --- | --- |
| **lat** | Lower latitude bound, if maxlat is present, otherwise latitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlat* | Upper latitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |
| **lon** | Lower longitude bound, if maxlon is present, otherwise longitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlon* | Upper longitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |

If maxlat and maxlon are not present, the area searched for locations will be determined by [lat – 0,01, lat + 0,01, lon – 0,01, lon + 0,01]. If both parameters are present, locations will be searched in the area determined by all four parameters.

This behaviour is the same in all the commands with latitude and longitude parameters.

### Return data

List of locations:

|  |  |  |
| --- | --- | --- |
| id | | ID of location |
| latitude | | Latitude of the location |
| longitude | | Longitude of the location |
| *altitude* | | *Optional:* Altitude of the location in meters |
| title | | Title of the location |
| *description* | | *Optional:* Description of the location |
| *dbpedia\_link* | | *Optional:* Link to a DBPedia-Page describing the location |
| created | | Date and time of creation |
| modified | | Date and time of modification |

# locations/**lat**/*maxlat*/**lon**/*maxlon*/stories/story\_count/

## GET

This retrieves all locations in a certain area, which have at least one story attached to it. This also includes the count of stories attached to the locations.

### URL-Parameters

|  |  |
| --- | --- |
| **lat** | Lower latitude bound, if maxlat is present, otherwise latitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlat* | Upper latitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |
| **lon** | Lower longitude bound, if maxlon is present, otherwise longitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlon* | Upper longitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |

### Return data

List of locations:

|  |  |  |
| --- | --- | --- |
| id | | ID of location |
| latitude | | Latitude of the location |
| longitude | | Longitude of the location |
| *altitude* | | *Optional:* Altitude of the location in meters |
| title | | Title of the location |
| *description* | | *Optional:* Description of the location |
| *dbpedia\_link* | | *Optional:* Link to a DBPedia-Page describing the location |
| created | | Date and time of creation |
| modified | | Date and time of modification |
| stories | | List of IDs of attached stories |

# locations/**lat**/*maxlat*/**lon**/*maxlon*/stories/title/

## GET

This retrieves all locations in a certain area, which have at least one story attached to it. This also includes the title of the stories attached to these locations.

### URL-Parameters

|  |  |
| --- | --- |
| **lat** | Lower latitude bound, if maxlat is present, otherwise latitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlat* | Upper latitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |
| **lon** | Lower longitude bound, if maxlon is present, otherwise longitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlon* | Upper longitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |

### Return data

List of locations:

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | abstract | Abstract of the story | | location | ID of the location | |

# locations/**lat**/*maxlat*/**lon**/*maxlon*/stories/

## GET

This retrieves all locations in a certain area, which have at least one story attached to it. This also all story information that is available.

### URL-Parameters

|  |  |
| --- | --- |
| **lat** | Lower latitude bound, if maxlat is present, otherwise latitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlat* | Upper latitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |
| **lon** | Lower longitude bound, if maxlon is present, otherwise longitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlon* | Upper longitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |











Return dataList of locations

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | location | ID of the location | | abstract | Abstract of the story | | author | ID of the user that created the story | | time\_start | Start point of story | | *time\_end* | *Optional:* End point of story | | *text* | *Optional:* Text of story | | created | Date and time of creation | | modified | Date and time of modification | | categories | List of IDs of attached categories | | |

# locations/**lat**/*maxlat*/**lon**/*maxlon*/stories/title/image/

## GET

This retrieves all locations in a certain area, which have at least one story attached to it. This also all story information that is available. This also includes the first image asset for each story, if present.

### URL-Parameters

|  |  |
| --- | --- |
| **lat** | Lower latitude bound, if maxlat is present, otherwise latitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlat* | Upper latitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |
| **lon** | Lower longitude bound, if maxlon is present, otherwise longitude around which locations will be returned  Format: [0-9]{1,3}.[0-9]{1,10} |
| *maxlon* | Upper longitude bound  Format: [0-9]{1,3}.[0-9]{1,10} |

### Return data

List of locations

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | abstract | Abstract of the story | | assets | List of assets:   |  |  | | --- | --- | | alt | Alternative text for first image | | sources | Source for first image | | |

# locations/**id**/

## GET

This retrieves information for one particular location.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be retrieved |



Return dataList of locations:

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |

## PUT

Authentication Required

This updates the respective location in the database.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be updated |

### Parameters

|  |  |
| --- | --- |
| *title* | *Optional:* Title of the location |
| *latitude* | *Optional:* Latitude of the location |
| *longitude* | *Optional:* Longitude of the location |

### Return data

Depending on whether the authentication succeeded or failed, either a 200 OK or a 403 Forbidden status code.

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| title | Title of the location |

# locations/**id**/stories/story\_count/

## GET

This retrieves information for one particular location. This also includes the count of stories attached to the location.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of IDs of attached stories |

# locations/**id**/stories/title/

## GET

This retrieves information for one particular location. This also includes the title of the stories attached to the location.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | abstract | Abstract of the story | | location | ID of the location | |

# locations/**id**/stories/

## GET

This retrieves information for one particular location. This also all story information that is available.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | location | ID of the location | | abstract | Abstract of the story | | author | ID of the user that created the story | | time\_start | Start point of story | | *time\_end* | *Optional:* End point of story | | *text* | *Optional:* Text of story | | created | Date and time of creation | | modified | Date and time of modification | | categories | List of IDs of attached categories | |

# locations/**id**/stories/title/image/

## GET

This retrieves information for one particular location. This also all story information that is available.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the location to be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of location |
| latitude | Latitude of the location |
| longitude | Longitude of the location |
| *altitude* | *Optional:* Altitude of the location in meters |
| title | Title of the location |
| *description* | *Optional:* Description of the location |
| *dbpedia\_link* | *Optional:* Link to a DBPedia-Page describing the location |
| created | Date and time of creation |
| modified | Date and time of modification |
| stories | List of stories:   |  |  | | --- | --- | | id | ID of the story | | title | Title of the story | | abstract | Abstract of the story | | assets | List of assets:   |  |  | | --- | --- | | alt | Alternative text for first image | | sources | Source for first image | | |

# stories/

## GET

This retrieves all the saved stories with or without a location.

### URL-Parameters

None

### Return data

List of stories:

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |

## POST

Authentication Required

This creates a new story in the database.

### Parameters

|  |  |
| --- | --- |
| **title** | Title of the story |
| **author** | Longitude of the location |
| *text* | Title of the story |
| **abstract** | Abstract of the story |
| **time\_start** | Startpoint of the story |
| *time\_end* | *Optional:* Endpoint of the story |
| **type** | Type of the story |
| *location* | *Optional:* ID of the location this story is attached to. |

### Return data

Depending on whether the authentication succeeded or failed, either a 201 Created or a 403 Forbidden status code. If authentication succeeds, the following header data will be returned.

|  |  |  |
| --- | --- | --- |
| Location | stories/**story\_id** | URI of the newly created story |

# stories/title/

## GET

This retrieves all the saved stories and their title.

### URL-Parameters

None

### Return data

|  |  |
| --- | --- |
| id | ID of story |
| title | Title of the story |
| *location* | *Optional:* ID of the location this story is attached to. |

# stories/**id**/

## GET

This retrieves information for one particular story.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the story to be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |
| assets | List of IDs of assets attached to this story |

## PUT

Authentication Required

This updates the respective story in the database.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the story to be updated |

### Parameters

|  |  |
| --- | --- |
| *title* | *Optional:* Title of the story |
| *text* | *Optional:* Text of the location |
| *time\_start* | *Optional:* Startpoint of the story |
| *time\_end* | *Optional:* Endpoint of the story |
| *category* | *Optional:* Type of the story |
| *location* | *Optional:* ID of the location this story is attached to. |

### Return data

Depending on whether the authentication succeeded or failed, either a 200 OK or a 403 Forbidden status code.

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |
| assets | List of IDs of assets attached to this story |

## DELETE

Authentication Required

This will delete the respective story.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the story to be deleted |

### Return data

Depending on whether the authentication succeeded or failed, either a 200 OK or a 403 Forbidden status code.

# stories/title/**query**/title/

## GET

This retrieves a list of stories where the title contains the query string. This can be used for auto-completion or similar.

### URL-Parameters

|  |  |
| --- | --- |
| **query** | Query string which should the story title should contain. |

### Return data

List of stories:

|  |  |
| --- | --- |
| id | ID of story |
| title | Title of the story |

# stories/text/**query**/title/

## GET

This retrieves a list of stories where the text or the title contains the query string. This can be used for auto-completion.

### URL-Parameters

|  |  |
| --- | --- |
| **query** | Query string which should the story text or title should contain. |

### Return data

List of stories:

|  |  |
| --- | --- |
| id | ID of story |
| title | Title of the story |

# stories/title/**query**/

## GET

This retrieves a list of stories where the title contains the query string. This can be used for search. This includes all the information of the stories.

### URL-Parameters

|  |  |
| --- | --- |
| **query** | Query string which should the story title should contain. |

### Return data

List of stories

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |
| assets | List of IDs of assets attached to this story |

# stories/text/**query**/

## GET

This retrieves a list of stories where the text or the title contains the query string. This can be used for search. This includes all the information of the stories.

### Parameters

|  |  |
| --- | --- |
| **query** | Query string which the story text or title should contain. |

### Return data

List of stories

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |
| assets | List of IDs of assets attached to this story |

# stories/**id**/image/

This retrieves one story and its included media source files.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the story, for which the media information should be retrieved |

### Return data

|  |  |
| --- | --- |
| id | ID of the story |
| title | Title of the story |
| location | ID of the location |
| abstract | Abstract of the story |
| author | ID of the user that created the story |
| time\_start | Start point of story |
| *time\_end* | *Optional:* End point of story |
| *text* | *Optional:* Text of story |
| created | Date and time of creation |
| modified | Date and time of modification |
| categories | List of IDs of attached categories |
| assets | List of assets:   |  |  | | --- | --- | | alt | Alternative text for first image | | sources | Source for first image | |

# stories/**id**/assets/

## GET

This retrieves all additional media that is attached to a particular entry.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the story, for which the media information should be retrieved |

### Return data

List of assets:

|  |  |
| --- | --- |
| type | “vid”, “img”, “txt”, or “aud” Determines whether this media file is a video, audio or image file |
| created | Date of creation |
| modified | Date of modification |
| alt | Alternative text for media file |
| description | Description of this asset |
| width | Width of media file if applicable |
| height | Height of media file if applicable |
| resolution | Resolution of media file if applicable |
| device | Device of media file if applicable |
| length | Length of media file if applicable |
| is\_readable | Determines whether the file is screen readable |
| sources | List of sources for media file:   |  |  | | --- | --- | | mime | MIME-Type of the source file | | source | URL to source file | |

## POST

Authentication Required

This creates a new asset for the respective story.

### Parameters

|  |  |
| --- | --- |
| **type** | Type of the media file, e.g. “vid”, “img”, or “aud” |
| **alt** | Alternative text for media file |
| *description* | *Optional:* description of the media file |
| *width* | *Optional:* Width of media file if applicable |
| *height* | *Optional:* Height of media file if applicable |
| *resolution* | *Optional:* Resolution of media file if applicable |
| *device* | *Optional:* Device of media file if applicable |
| *length* | *Optional:* Length of media file if applicable |
| *is\_readable* | *Optional:* Determines whether the file is screen readable |

### Return data

Depending on whether the authentication succeeded or failed, either a 201 Created or a 403 Forbidden status code. If authentication succeeds, the following header data will be returned.

|  |  |  |
| --- | --- | --- |
| Location | assets/**asset\_id** | URI of the newly created asset |

# assets/

## GET

This lists all assets.

### Return data

List of assets:

|  |  |
| --- | --- |
| type | “vid”, “img”, “txt”, or “aud” Determines whether this media file is a video, audio or image file |
| created | Date of creation |
| modified | Date of modification |
| alt | Alternative text for media file |
| description | Description of this asset |
| width | Width of media file if applicable |
| height | Height of media file if applicable |
| resolution | Resolution of media file if applicable |
| device | Device of media file if applicable |
| length | Length of media file if applicable |
| is\_readable | Determines whether the file is screen readable |
| sources | List of sources for media file:   |  |  | | --- | --- | | mime | MIME-Type of the source file | | source | URL to source file | |

## POST

Authentication Required

This creates a new asset without a story attached to it.

### Parameters

|  |  |
| --- | --- |
| **type** | Type of the media file, e.g. “vid”, “img”, or “aud” |
| **alt** | Alternative text for media file |
| *description* | *Optional:* description of the media file |
| *width* | *Optional:* Width of media file if applicable |
| *height* | *Optional:* Height of media file if applicable |
| *resolution* | *Optional:* Resolution of media file if applicable |
| *device* | *Optional:* Device of media file if applicable |
| *length* | *Optional:* Length of media file if applicable |
| *is\_readable* | *Optional:* Determines whether the file is screen readable |

### Return data

Depending on whether the authentication succeeded or failed, either a 201 Created or a 403 Forbidden status code. If authentication succeeds, the following header data will be returned.

|  |  |  |
| --- | --- | --- |
| Location | assets/**asset\_id** | URI of the newly created asset |

# assets/**asset\_id**

## GET

This retrieves one specific media file that is attached to the story.

### URL-Parameters

|  |  |
| --- | --- |
| **asset\_id** | ID of the media entry |

### Additional Request Headers

For this method, the Accept header must be set. The server will try to serve the file matching the requested MIME-Type. If this is not possible, a 406 Not Acceptable status code will be sent, followed by the accepted Content-Types.

|  |  |
| --- | --- |
| **Accept** | Desired MIME-Type of the media file. |

## PUT

Authentication Required

This updates the respective asset.

### Parameters

|  |  |
| --- | --- |
| **type** | Type of the media file, e.g. “vid”, “img”, “txt”, or “aud” |
| **alt** | Alternative text for media file |
| *description* | *Optional:* description of the media file |
| *width* | *Optional:* Width of media file if applicable |
| *height* | *Optional:* Height of media file if applicable |
| *resolution* | *Optional:* Resolution of media file if applicable |
| *device* | *Optional:* Device of media file if applicable |
| *length* | *Optional:* Length of media file if applicable |
| *is\_readable* | *Optional:* Determines whether the file is screen readable |

### Return data

Depending on whether the authentication succeeded or failed, either a 200 OK or a 403 Forbidden status code.

## DELETE

Authentication Required

This will delete the media entry and all of its sources.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the media entry to be deleted |

### Return data

Depending on whether the authentication succeeded or failed, either a 200 OK or a 403 Forbidden status code.

# assets/**asset\_id**/sources

## PUT

Authentication Required

This uploads a new media source to this asset.

### Parameters

|  |  |
| --- | --- |
| mime | The MIME-Type of the file |
| file | The actual media file |

# User authentication

The following section describes methods for basic user authentication. In order to post or update stories, the source of the information needs to be authenticated. This can be done in two ways:

1. User authentication  
     
   Choosing this method, users will have to register and log-in before they can create new stories. The following methods regarding **sessions** and **users** are used for this. After having logged-in the session ID can be used as an authentication token for posting or updating stories. Authentication then is done by sending an additional request header field containing the authentication token like this:

Authorization: Token 9944b09199c62bcf9418ad846dd0e4bbdfc6ee4b  
  
Additionally, session authentication mode can be used. The session ID will then be saved as a cookie on the client. Since this violates REST principles, it is not used by default and has to be indicated by setting the **session**-parameter to **True** when logging in.  
  
Note that these authentication modes need to be used for every request marked with Authentication Required. Otherwise a 401 Unauthorized status code will be returned.

Please note, that HTTPS should always be used when using this method, since the Token and the user credentials will be transmitted unencrypted otherwise.  
  
User registration is handled using the default Django admin backend.

1. Moderated mode  
   If this mode is active, users will not have to register and log-in before posting stories. Instead, new stories posted by the users will be assigned a temporary ID so they can be updated later on and as soon as they are finished, they are sent to the backend editing system. Here they can be moderated and put into the database.

The currently active authentication mode is determined by the setting AUTHENTICATION\_MODE in the local\_settings.py.

|  |  |
| --- | --- |
| **user\_authentication** | User authentication mode is active |
| **moderation** | Moderated mode is active |

# sessions/

## POST

This is used to create a new logged-in session for a user registered in the database. It will return a session key, which can be used later to create new locations and stories.

### URL-Parameters

None

### Parameters

|  |  |
| --- | --- |
| **user** | Username |
| **password** | Password |
| *session* | *Optional:* True or False, depending on whether the session should be saved as a cookie. Defaults to False. |

### Return data

Depending on whether the authentication succeeded or failed, either a 201 Created or a 403 Forbidden status code. If authentication succeeds, the following header data will be returned.

|  |  |  |
| --- | --- | --- |
| Location | sessions/**session\_id** | Location of the newly created session |

If authentication fails, an ID and a message will be sent, explaining why the authentication failed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| fail\_id | ID of reason the authentication failed   |  |  | | --- | --- | | 1 | Wrong credentials supplied | | 2 | User is banned for posting malicious content | |
| message | Message explaining the ID |

# sessions/**id**/

Authentication Required

## GET

This will retrieve the user details of the currently logged in user. The user ID can be used to change the password or other user details.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the session |

### Return data

|  |  |
| --- | --- |
| user\_id | ID of the logged in user |
| user\_name | User name of the logged in user |
| first\_name | First name of the user |
| last\_name | Last name of the user |
| email | E-mail address of the user |

## DELETE

This is used to destroy the respective session and to log-out the user.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the session to be destroyed. |

### Return data

If the session was found, a 204 No Content status code will be returned. If the session was not found, a 404 Not Found status code will be returned.

# users/

## POST

This is used to register a new user. The user will not be logged in unless a new session is requested as stated above.

### URL-Parameters

None

### Parameters

|  |  |
| --- | --- |
| **first\_name** | First name of the user |
| **last\_name** | Last name of the user |
| **user\_name** | User name |
| **email** | E-mail address of the user |
| **password** | Password |

### Return data

If the user was created successfully, a 201 Created status code will be returned. The following header data will be sent. If the user could not be created, a 500 Internal Server Error will be returned.

|  |  |  |
| --- | --- | --- |
| Location | users/**user\_id** | Location of the newly created user |

# users/**id**/

### Authentication Required

## PUT

This is used to update the user profile. Most of the parameters are optional, only the user id and password are required at all times.

### URL-Parameters

|  |  |
| --- | --- |
| **id** | ID of the user to be updated. |

### Parameters

|  |  |
| --- | --- |
| *first\_name* | First name of the user |
| *last\_name* | Last name of the user |
| **user\_name** | User name |
| *email* | E-mail address of the user |
| **password** | Password |
| *new\_password* | New password to be set for the user |

### Return data

If the user could be authenticated, then the server will update the user profile and return a 204 No Content status code. If the user could not be authenticated, a 401 Unauthorized will be returned.

# Moderated mode

When using the moderated mode, stories will not be posted to the database directly. Instead they are displayed in the backend editing system, where they can be moderated and then put in the database. However, some of the APIs should still be usable without user authentication, for example creating a new location or updating a previously entered own story. For this, Django’s Cross Site Request Forgery mechanism will be used. See more here: <https://docs.djangoproject.com/en/1.6/ref/contrib/csrf/>.