

Session



If the API changes in an incompatible manner this is listed explicitly in the "important release notes" section of each [releases changelog](#).

The session API is located in the editor module, it enables authentication to translate5 via API.

Provides actions about the authenticated user

User sessions can be created / deleted by this API. Also some status information to the authenticated user can be retrieved.

URL:	/editor/session/[SESSION_ID]
Available Methods:	GET/POST/DELETE
Specialities:	See below

POST Request

The POST requests needs the following two parameters either as data object or plain form parameters:

Name	Type	Info
login	string	The user to be authenticated.
passwd	string	The passwd of the user to be authenticated.
taskGuid	string	Optional and obsolete: The taskGuid of the task which should be opened for the user. Obsolete since accessing and opening tasks is now controlled by the URL. So just navigate to /editor/taskid/123/

POST resulting resource layout

Returns HTTP 200 when login was successful, 400 when parameters are missing, 403 when credentials were wrong.
When using the optional parameter taskGuid, 404 can happen when the task to the given taskGuid can not be found. When the user is no PM user and it is not associated to the task, also a 403 is triggered.

A successful result looks like:

Name	Type	Info
sessionId	string	the sessionId of the newly created session, can be used directly for further API communication. The given sessionId is essential for further usage of other requests to the API , see below in the section API Usage .
sessionToken	string	A one time usable token which can be passed to the users browser for overtaking the created session. For example: Users are administered by your software, translate5 is just embedded. Users authenticate themselves on your system, your system authenticates them via API at translate5. Your system then provides a link containing the sessionToken parameter: /editor?sessionToken=RECEIVED_TOKEN After clicking on this URL translate5 will be opened with the authenticated user. The token is only callable one time to prohibit misuse.
taskUrlPath	string	The URL path part to the task, if in POST request a taskGuid was provided.

GET resulting resource layout

For GET the sessionId must be part of the URL, like the usual ID parameter: /editor/session/SESSION_ID

Returns HTTP 200 when user is authenticated, 404 otherwise.

Name	Type	Info
------	------	------

state	string	contains "authenticated" or "not authenticated"
user	object	contains the authenticated user or null. The users structure is as described in the user API .

DELETE call

For DELETE the sessionId must be part of the URL, like the usual ID parameter: /editor/session/SESSION_ID

Returns HTTP 200 when session was destroyed, 404 otherwise.

Instead of the session ID also the internalSessionUniqId can be provided. It then has to start with the prefix "internal-": DELETE /editor/session/internal-SESSION_ID

That's possible for DELETE only, and the calling user must be an API user.

Special session api calls

URL:	/editor/session/impersonate
Available Methods:	GET
Parameters:	login
Specialities:	Impersonate/log-in as different user in translate5. The "user to impersonate" is defined as GET "login" parameter (ex: /editor/session/impersonate?login=manager4). Once you are authenticate/impersonate user with no api roles, you can no longer use this action (because of api only users acl restriction).
Requirements:	Authenticate user with api role is required. Also the "login" parameter is mandatory.

Using the received sessionId in other API calls

Translate5 does not use a traditional RESTful authentication, but uses sessions for this purpose.

To use the API as an authenticated user, first a session must be generated by the above mentioned session POST call.

The received sessionId must be given on each other request to the API as cookie HTTP header.

Example HTTP Header to provide the Session ID as cookie value:

```
Cookie: zfExtended=GIVEN_SESSION_ID
```