

Task

- [Provides access to the tasks stored in translate5.](#)
- [Special Task Import URLs](#)
- [Special Task Export URLs](#)
- [Other Special Task URLs](#)
- [Task Resource Layout](#)
- [Import push callback](#)
- [Task types](#)



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

Provides access to the tasks stored in translate5.

URL:	/editor/task/[ID]
Available Methods:	GET / POST / PUT / DELETE
Specialities:	<p>Since on a POST request the import data is given, the POST must be transmitted as "multipart/form-data".</p> <p>POST creates the Task, and start the import process. Unless autoStartImport is set to 0, then the task is uploaded but the import starts on calling or a separate URL is called. See below.</p> <p>By setting the userState to the value "edit" the task is registered as to be edited by the current user.</p> <p>Setting it back to the userState "open" means that the the task is open again for all users.</p> <p>forced DELETE: if the ACL right "taskForceDelete" (resource "backend") is set to a role, users of that role can use the "force" switch to delete tasks locked by other users.</p> <p>This makes sense in some environments where translate5 task administration is done via API only. A DELETE call would then look like: DELETE /editor/task/123?force=1</p>
Advanced filtering	<p>In addition to the default filtering possibilities by task model fields, there is additional option for advanced "workflow" filtering. For example you can filter for all tasks where user "X" is assigned as reviewer. The additional filter parameter should look like this:</p> <pre>filter: [{"operator": "in", "value": [{"4484fa0b-4c94-4294-8d30-7dd697760d2b"}], "property": "userName"}, {"operator": "in", "value": ["reviewer"], "property": "workflowUserRole"}]</pre> <p>This will filter for all tasks where the user with guid {4484fa0b-4c94-4294-8d30-7dd697760d2b} is assigned as reviewer.</p> <p>The other available fields for the advanced workflow filters:</p> <ul style="list-style-type: none">• userName: filters for assigned user in the workflow by user guid• userState: filters for assigned workflow state (role)• assignmentDate: filter for user job assignment date• finishedDate: filter for user job finish date• deadlineDate: filter for job deadline date

Special Task Import URLs

URL to start the import of the uploaded task:	/editor/task/[ID]/import [ID] is the id of the task, the taskGuid is currently not supported
Reimport a before exported task from an excel file:	editor/task/[ID]/excelreimport

Special Task Export URLs

URL to get the edited task data in a ZIP container:	<p>/editor/task/export/id/[ID]</p> <p>The export ZIP may contain the translated content directly, or for legacy reasons in a folder named as the taskGuid of the task.</p> <p>This can be controlled by setting runtimeOptions.editor.export.taskguiddirectory or explicitly on each export call by setting a GET parameter "taskguiddirectory" with 0 or 1. example: /editor/task/export/id/[ID]?taskguiddirectory=0 to explicitly remove the guid, or 1 to have it in the folder.</p>
As above, but with diff tags:	/editor/task/export/id/[ID]/diff/1
Export the translated file only from the ZIP container:	/editor/task/export/id/[ID]?format=filetranslation
MQM Statistic Export:	/editor/qmstatistics/index/taskGuid/[GUID]?type=[TYPE]
Export Task as Excel:	/editor/task/[ID]/excelexport/

Where [ID] is the value of the mysql-id field, [GUID] the mysql-taskGuid and [TYPE] a valid field type of the task (field type refers to the column the MQM has been used on in the editor).

Other Special Task URLs

URL to trigger specific workflow actions to that task via API.	<p>POST /editor/task/[ID]/workflow [ID] is the id of the task, the taskGuid is currently not supported</p> <p>Parameter: "trigger" which contains a string, for example: "notifyAllUsersAboutTaskAssociation"</p> <p>The workflow action must be configured in the LEK_workflow_action table.</p> <p>See also Workflow Actions and examples.</p>
Show task import progress:	<p>editor/task/importprogress?taskGuid=[GUID]</p> <div> <p>Resource layout example</p> <pre>{ "progress": { "progress": 99.58, // total import progress in percent "workersDone": 10, // total task workers with state done "workersTotal": 11, // total queued task workers "taskGuid": [GUID] // taskGuid "workerRunning": "editor_Plugins_SegmentStatistics_WriteStatisticsWorker" } }</pre> </div>

Task Resource Layout

Name	Type	Info
id	int	GET only
taskGuid	string	GET only
entityVersion	integer	GET only, set by server
modified	date	GET only, contains the last modification timestamp of the task entity
taskNr	string	
foreignId	string	optional, is a 120 character long freetext field, not shown in the GUI. Can be used in installations where translate5 is used embedded only, to track the taskId of the management system for example.
taskName	string	

description	string	optional, max 500 character long textual description of the project / task
foreignName	string	optional, is a 255 character long, see foreignId.
sourceLang	string /integer	POST accepts: integers (internal ID), rfc5646 string values, LCIDs (must be prefixed with "Icid-" for example Icid-1031), GET returns always the internally used language Id
targetLang	string /integer /array	POST accepts: integers (internal ID), rfc5646 string values, LCIDs (must be prefixed with "Icid-" for example Icid-1031), GET returns always the internally used language Id. If array of target languages is provided, for each target language, separate task (project task) will be created
relaisLang	string /integer	POST accepts: integers (internal ID), rfc5646 string values, LCIDs (must be prefixed with "Icid-" for example Icid-1031), GET returns always the internally used language Id
locked	date	GET only
lockingUser	string	GET only
lockingUserName	string	GET only. When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
lockedInternalSessionUniqId	string	GET only
state	string	POST: can be omitted, is set automatically to "import" PUT: "end" to end a task , "open" to reopen an ended task. GET: returns the current state of the task import task is still importing data error task import produced and error and task is not usable end task is ended by a user open task is usable by users unconfirmed task is unconfirmed, can not be edited, must be confirmed (status open) first
workflow	string	The name of the workflow used in the task
workflowStep	integer	The workflow step nr
workflowStepName	string	The name of the current workflow step
pmGuid	string	
pmName	string	
wordCount	integer	
targetDeliveryDate (deprecated)	date	The date field is deprecated. Use LEK_taskUserAssoc deadlineDate instead.
referenceFiles	boolean	GET only
terminologie	boolean	GET only
orderdate	datetime	
enddate	datetime	GET only, contains the datetime when task is set to status end (calculated)
edit100PercentMatch	boolean	POST only, defines whether 100% matches found should be editable or locked
lockLocked	boolean	POST only, only for SDLXLIFF files: Segments explicitly locked in imported file are also locked in translate5
emptyTargets	boolean	GET only, flag if task has contained only empty targets on import time and is therefore a translation job, or otherwise contained some translations and is a review job
importAppVersion	string	GET only, contains the translate5 version used at import time
customerId	integer	The client for the task (= id from table LEK_customer)

enableSourceEditing	boolean	POST only
autoStartImport	boolean	POST only, optional, defaults to true. set it to false to prevent an automatic start of the import - for additional configuration, preprocessing before the real import.
qmSubEnabled	boolean	GET only
qmSubFlags	auto	GET only
qmSubSeverities	auto	GET only
userState	string	As defined by the workflow.
userRole	string	GET only
userStep	string	GET only
segmentFields	auto	GET only, list with associated segment fields. When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
userPrefs	auto	GET only, list with associated default user preferences. When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
users	auto	GET only, list with associated users
taskassocs	auto	With Language Resources : GET only, list with associated Language Resources. When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
userCount	integer	GET only
fileCount	integer	GET only When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
exportRunning	string	GET only
defaultSegmentLayout	boolean	GET only When requesting all tasks (GET /editor/task) this field is only delivered if a filter or limit parameter is provided.
importUpload	HTTP Upload	POST only
usageMode	string	Currently 'competitive' or 'cooperative' or 'simultaneous'. competitive: only the first user accepting the task may work on it. cooperative: All associated users may work on the task, but always one user at a time. simultaneous: All associated users may work on the same task at the same time. Note: if no value is provided on task creation, the value will be evaluated from this config runtimeOptions.import.initialTaskUsageMode This config is overridable on client level
segmentFinishCount	integer	GET only, calculated(reviewed/translated depends on the workflow step) number of segments in the current workflow
segmentCount	integer	GET only, total number of segments in the current task
workflowProgressSummary	string	GET only; Workflow step progress for each workflow step. Response layout: [{workflowStep: translation, status: finished, progress: 100}, {workflowStep: proofreading, status: running, progress: 33}, {workflowStep: translatorCheck, status: open, progress: 0}]
visualReview	file	POST only, Plugin Visual Review only. Adds a (single) file as the source for the visual review. If the import zip already contains a "visualReview" folder with review sources this file will be added to the ones provided in the zip. Filetype can be "pdf" or "html"

visualReviewHtml	array	<p>POST only, Plugin Visual Review only.</p> <p>An URL to a website that will be taken as a source(more info about the source linking here: Visual - WYSIWYG: Upload rules for layout sources) for the visual review.The URL will be downloaded with all linked resources. Overwrites any sources provided via "visualReview" or in the import package.</p> <p>The provided URL has to be fully qualified like "https://www.translate5.net/en/translate5-open-source-translation-system/" and has to be accessible without login etc.</p>
mappingType	string	<p>POST only, Plugin Visual Review only.</p> <p>Defines, if the visual review is mapped to the source or target of the segments. Defaults to "source". (source target)</p>
taskType	string	<p>GET only. Possible values:</p> <ul style="list-style-type: none"> ▪ default - default task type ▪ projectTask - task type when the task is part of project task collection ▪ project - root task for all project tasks ▪ instanttranslate-pre-translate - task type for translated files via instant-translate, is introduced via instant-translate plugin. <p>See below section task types for details.</p>
projectId	integer	<p>GET only, project id of the task. For taskType default,project and instanttranslate-pre-translate this will have the same value as the id. For tasks with taskType projectTask, this will have the value of the root task(task with taskType project) of the project task collection</p>
importProgress	integer	<p>GET only (the task must be in state import or matchanalysis), Current import(or running matchanalysis worker) total progress.</p>

Import push callback

To prevent excessive polling of the task state while import a HTTP callback can be configured. This is needed since task imports may need up to hours, depending on task size and used features (like term tagging).

To configure the callback search in the configuration for "runtimeOptions.import.callbackUrl" and set the value to your URL:

- If a URL is set, this URL is called with HTTP POST after task import, where the task itself is contained as JSON (only the meta data, not the uploaded files).
- The callback is also called in case of import errors, so your callback handler must evaluate the tasks state field given in json. A state of "open" (or in other words not "error") means a successful import of that task.
- If the callback is not returning a HTTP 200 status, translate5 will add a warning into the application log on each call of the callback.

Example JSON which is send as raw data to the configured callback endpoint

```
{
  "id": "2636",
  "entityVersion": "47",
  "modified": "2023-11-07 14:15:01",
  "taskGuid": "{8cdb06ec-840a-46e8-ad0a-5b44ff7d6197}",
  "taskNr": "",
  "foreignId": "",
  "taskName": "Translate5 Demo Text EN - en \\/ sq",
  "foreignName": "",
  "sourceLang": "5",
  "targetLang": "274",
  "relaisLang": "0",
  "locked": null,
  "lockingUser": null,
  "state": "open",
  "workflow": "default",
  "workflowStep": "1",
  "workflowStepName": "no workflow",
  "pmGuid": "{e6828cdf-2ee0-4a25-af0a-92e6f060e9eb}",
  "pmName": "Manager, Project (manager)",
  "wordCount": "92",
  "userCount": "0",
  "referenceFiles": "0",
  "terminologie": "0",
  "orderdate": "2023-10-31 00:00:00",
  "enddate": null,
  "enableSourceEditing": "0",
  "edit100PercentMatch": "0",
  "lockLocked": "1",
  "emptyTargets": "1",
  "importAppVersion": "development",
  "customerId": "1",
  "usageMode": "simultaneous",
  "segmentCount": "24",
  "segmentFinishCount": "0",
  "taskType": "projectTask",
  "projectId": "2565",
  "diffExportUsable": "0",
  "reimportable": "1",
  "description": "",
  "created": "2023-11-07 14:14:49"
}
```

The callback is called also in the case of an import error!

In seldom cases the import is crashing in a way no callback can be send (server reboot for example). In that case such a task is set to state error after 48 hours, which then triggers the callback finally. The 48 hours can be configured too: `runtimeOptions.import.timeout`

Task types

A task can be of a specific type. The types defines the visibility and how the task is threatened in the application. For example the projects are also tasks, but with `taskType` project.

The task type is always calculated / set in the application automatically and must not be set from outside.

Available task types:

- **default** - default task type, used for legacy tasks before introducing the project features, and for projects containing only one target language, in other words where `project = task`.
- **projectTask** - task type when the task is part of project task collection, for each target language one `projectTask` is created
- **project** - root task for all project tasks, listed in the project overview. Is not editable directly, only the corresponding `projectTasks` are editable.
- **instanttranslate-pre-translate** - task type for translated files via instant-translate, is introduced via instant-translate plugin. Tasks of this type are only used internally for translating an uploaded file via `instanttranslation`, they are not listed in ordinary `translate5` task and project listing.

