

InstantTranslate



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

Description

InstantTranslate is translate5 api interface which can be used for translating text and files using the translate5 language resources (term collections, translation memories or machine translation engines).

Available language resource for InstantTranslate are the assigned resources of a customers of the current logged-in user.

For translating files, the [file-extension must be configured](#) accordingly.

Translate text with translate5 language resources .

URL:	/editor/instanttranslateapi/translate
Available Methods:	POST
Specialities:	To be able to search for a text, there must be available language resources assigned to a customers of a user
Availability:	-

Translate text resource Layout

Name	Type	Info
text	string	The string to translate.
source	string	The source language to translate from. Must be valid rfc5646 code.
target	string	The target language to translate to. Must be valid rfc5646 code.
langresType	string	optional, one (or more, comma-seperated) language-resource types. e.g. "DeepL" or "DeepL,Google" The types can be ('TermCollection', 'NEC-TM', 'OpenTM2', 'DeepL', 'SDLLanguageCloud', 'GroupShare', 'Google', 'Moses', 'Lucy LT', 'TildeMT')
langresId	int	optional, one (or more, comma-seperated) language-resource id's, e.g "12345" or "12345,23456"
nosegmentation	int	optional, if set to "1" the string to translate will not be segmented (even if it contains punctuation). Please note, the result-layout with segmented & unsegmented texts is different
escape	int	optional, if set to "1" the returned texts and any markup will be escaped

Translate a file with translate5 language resources

Please ensure that:

- the filetype is supported (e.g. by Okapi) and
- LanguageResources for the desired language-combination are available (order of their application: first termCollections, second TMs, third MTs).

Translating files can take a while.

URL	Methods	Parameters	Description
-----	---------	------------	-------------

<p>/editor /instanttranslate api /filepretranslate</p>	<p>P O S T</p>	<p>source: string; the source language to translate from. Must be valid rfc5646 code.</p> <p>target: string; the target language to translate to. Must be valid rfc5646 code.</p> <p>file: multipart/form-data; the file to translate. "0" as field name is deprecated and will not be supported in one of the next transalte5 versions.</p> <p>OR (since 7.5.0)</p> <p>fileName: file name given as string in the POST parameter</p> <p>fileData: file content given as string in the POST parameter</p>	<p>Step 1: Starts a task-import with the file and its pretranslation.</p> <p>As long as the import is running, the response in step 2 for this file will say "isImporting" instead of returning the download-link. You might need to repeat step 2 until you get the link.</p> <p>The translated files are available until their configured lifetime is over; then they will be removed automatically.</p> <p>Response (example): {"taskId":"539"}</p>
<p>/editor/instanttr anslateapi /filelist</p>	<p>G E T</p>		<p>Step 2: Get a list of all pretranslations that are currently available for the InstantTranslate-url including their downloadUrl.</p> <p>Response: allPretranslatedFiles for tasks with taskId as key .</p> <div data-bbox="763 688 1485 1606" style="border: 1px solid #ccc; padding: 10px;"> <p>Response layout example</p> <pre> { "allPretranslatedFiles": { "2237": { "taskId": "2237", "taskName": "Apache short-en-de.html", "sourceLang": "en", "targetLang": "de", "downloadUrl": "/editor/task/export/id/2237?format=filetranslation", "removeDate": "2021-03-18", "importProgress": [] } "2464": { "taskId": "2464", "taskName": "Single word-de-en.html", "sourceLang": "de", "targetLang": "en", "downloadUrl": "isImporting", "removeDate": "2021-04-08", "importProgress": { "progress": 2, // import percentage "workersDone": 1, // total workers done "workersTotal": 9, // total queued workers "taskGuid": "{a5945d9f-4007-4063-ac17-41da3003dab1}", "workerRunning": "editor_Plugins_Okapi_Worker" } } }, "dateAsOf": "2021-04-06 17:36:42" } </pre> </div> <ul style="list-style-type: none"> downloadUrl: 'isImporting' 'isErroneous' 'isNotPretranslated' url (the url starts a common task-export; the translated file is included in the exported ZIP) importProgress: check Task importprogress api

<p>editor /instanttranslate api/writetm</p>	<p>P O S T</p>	<p>The request parameters must be encapsulated in data parameter as json.</p> <p>ex:</p> <pre>data={"sourceLanguage": "en","targetLanguage": "de","source": "Simple text","target": "Einfacher Text"}</pre> <p>An example call via CURL could look like this:</p> <pre>curl --location --request POST 'http://t5docker.localdev/editor /instanttranslateapi/writetm' \ --header 'Content-Type: application/x- www-form-urlencoded; charset=UTF-8' \ --header 'Accept: application/json' \ --header 'Translate5AuthToken: {{token}}' \ --form 'data="{\"sourceLanguage\": \" en\", \"targetLanguage\": \"de\", \"source\": \"Simple text hello\", \" target\": \"Einfacher Text hallo\"}'</pre>	<h3>Auto creation of memory</h3> <p>For each customer and language and language combination provided in the request(see example on the left), one memory will be auto created. And the source and the target content will be saved in those memories.</p> <p>Which customers will be used? The assigned customers to all users, not the current user which is making the request, but all available users in translate5 with instantTranslate role.</p> <p>With the runtimeOptions.InstantTranslate.saveToServices (overwrite on client level) it can be defined which TM memory resources can be used. By default this is disabled on instance level. Currently only available options are t5memory.</p>
<p>/editor /instanttranslate api /filepretranslate now (since 7.15.0)</p>	<p>P O S T</p>	<p>source: string; the source language to translate from. Must be valid rfc5646 code.</p> <p>target: string; the target language to translate to. Must be valid rfc5646 code.</p> <p>file: multipart/form-data; the file to translate. "0" as field name is deprecated and will not be supported in one of the next transalte5 versions.</p> <p>OR</p> <p>fileName: file name given as string in the POST parameter</p> <p>fileData: file content given as string in the POST parameter</p>	<p>Starts a task-import with the file and its pretranslation.</p> <p>Waits until the import is finished and directly returns the results of the translated task.</p> <p>The translated files are available until their configured lifetime is over; then they will be removed automatically.</p> <p>Response examples: {"taskId":"539","fileTranslation":".."} {"taskId":"539","error":"Task import timeout exceeded (60s)"}</p>

Configuration of InstantTranslate

<p>runtimeOptions.LanguageResources. searchCharacterLimit</p>	<p>m ap</p>	<p>Maximum character per language resource allowed for search. The configuration key is the language resource id, and the value is the character limit. Ex: {"1": 100},{2": 300}</p>
<p>runtimeOptions.InstantTranslate. pretranslationTaskLifetimeDays</p>	<p>int 2</p>	<p>How many days do you want to keep pretranslated files before the system removes them?</p>
<p>runtimeOptions.InstantTranslate. saveToServices</p>	<p>st ri ng</p>	<p>If value other than "disabled" is selected, instantTranslate translations can be saved in separate TM of the selected type (s) by calling editor/instanttranslateapi/writetm. Currently only OpenTM2 is supported.</p>
<p>runtimeOptions.InstantTranslate.user. defaultLanguages</p>	<p>m ap 1</p>	<p>Default search languages for instant-translate which will be pre-selected when instant-translate is loaded.</p> <p>Example how to configure the config:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Config value</p> <pre>{ "sourceLangDefault": "en", "targetLangDefault": "de" }</pre> </div>