Globalese API v2.1
Developers' Guide

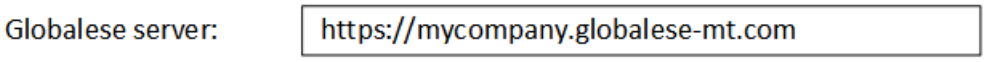
Globalese

Global settings

Depending on the usage scenario, three settings can possibly be used globally:

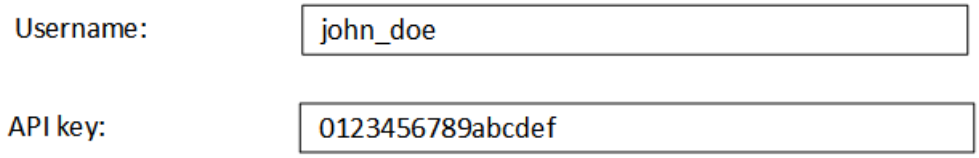
- Globalese server URL,
- Username and API key

1. **Globalese server**



When using v2.1 of the Globalese API, the API base URL is the Globalese instance URL followed by `/api/v2.1`. In this example, you need to prefix all API endpoints with `https://mycompany.globalese-mt.com/api/v2.1`

2. **Globalese username and API key**



API keys are tied to user accounts. The API key of the user is not the same as the password used for signing in to the Globalese portal.

A user can change their API key any time through the Globalese portal.

Checking global settings

Once the user has provided the Globalese server URL, their username and API key, a quick check may be run against the Globalese API to see whether the data provided can authenticate the user. Please see “Checking user credentials” in the API documentation.

Project-specific settings

We **strongly** suggest grouping files from the same project in a corresponding Globalese project. The Globalese project should ideally exist throughout the lifetime of the source (or CAT tool) project, to allow any newly added or re-imported files to be processed in the same Globalese project.

We do **not** recommend creating one project per file sent to Globalese if the files are in the same project at the source, as this creates unnecessary overhead and extra API calls.

Restrictions when choosing an engine for a project:

- A Globalese project may only be created in a group that the Globalese user authenticated in the API call is member of.
- The choice of engines that can be used for a project is limited to engines in the same group as the project.
- A Globalese project can only have one engine assigned to it. The engine may be changed for an existing project, although this should not be necessary.

Project settings that should be automatically taken from CAT tool project properties

The user should not be prompted for the following attributes:

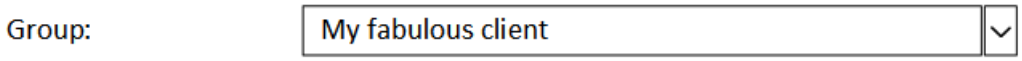
- Project name
- Project source language
- Project target language

Project settings that must be provided by the user for each project

1. **Group**

See `GET /groups`

We suggest presenting groups as a dropdown list.



The list of groups offered to the user is simply the list of groups obtained from the Globalese API. The result from that call only includes the groups the user is a member of.

2. **Engine**

See `GET /engines`

We suggest presenting engines as a dropdown list.

The list of engines should be presented to the user **only after** the group has been selected. The filtering parameters for the query should include the group, the source and the target language.

Note: The status of the engines in the list is important, and therefore it might be useful to show it to the user. Depending on the usage scenario, it may or may not make sense to filter for only those engines that can translate immediately (i.e. are not being trained or have not even been trained). Use the `ready` property of the `GET /engines` response to determine whether an engine is ready for translation.

Recommended workflow for processing files in a project

0. Store global settings first

The user should not be prompted for the global settings for every project they work on. Instead, if the global settings have not been provided when a user wants to process files in Globalese, the user should be prompted to provide the global settings first.

1. Create Globalese project

See `POST /projects`

As described above. The API returns the new Globalese project ID, which should be stored along with the project for later use, as long as the local (CAT tool) project exists. This will make it easier to process files that are added later or re-imported with different settings to the local (CAT tool) project.

2. Transfer files to Globalese project

See `POST /translation-files`
and `POST /translation-files/{id}`

Since the settings for creating a Globalese project are specific to the local (CAT tool) project, no further user input is required, apart from a selection of files.

`POST /translation-files` returns the Globalese file ID for a new file. This ID should be stored and used for uploading the file (using `POST /translation-files/{id}`), as well as for tracking the file status and downloading the translated file later.

3. Trigger translation

See `POST /translation-files/{id}/translate`

We **strongly** suggest triggering the translation of all files in the project in one batch, i.e. sending the above request for each file one after the other, one at a time, immediately after receiving a response from the previous request.

We do **not** recommend triggering the translation of subsequent files only after the previous file has been translated and downloaded. As the API interaction is stateless, Globalese will consider the project finished after the first file is finished. Consequently, when the translation of the second file is triggered, Globalese will need to deploy the same engine again for the second file, resulting in considerable and unnecessary ramp-up times.

4. Wait and poll project status / file statuses regularly

See `GET /projects/{id}`
and `GET /translation-files/{id}`

When translating an entire project, it is enough to call `GET /projects/{id}`. If it is important to track individual files, make a call to `GET /translation-files/{id}` using the individual file IDs.

5. Retrieve machine-translated files

See `GET /translation-files/{id}/download`

A file may be downloaded when its status changes to `translated`.

6. Clean up

See `DELETE /projects/{id}`
and `DELETE /translation-files/{id}`

If a file is re-imported to the project, it is recommended to delete the old version of the file in the Globalese project and re-upload the file.

When a file/project is deleted locally (in the CAT tool), the associated Globalese file/project should also be deleted to prevent it from existing in Globalese forever.

Document version number	Last modified on	Last modified by
1.0	2014-09-30	Gergely Horváth
1.1	2016-07-27	Gergely Horváth
1.2	2016-10-10	Gergely Horváth
2.0	2017-05-08	Gergely Horváth
2.1	2019-01-24	Gergely Horváth