How to Guide to set up a Rigi project

Version 0.4 (27 April 2020)

Rigi.io - Localization Platform

Table of contents

Getting started

This guide describes how to set up a Rigi project on the Rigi server.

Jump directly to step-by-step instructions.

Intended audience

- Developer
- Project Manager

Purpose

- Set up a Rigi project and ensure that the following becomes available:
    - Source and target files with source texts and/or translations.
    - Source texts
    - Target texts (if available)
    - HTML Previews

Prerequisites

- You need to have permission to create a Rigi project on the Rigi server.

References

- How to Guide to capture HTML Previews.
- How to upload texts using the Translation Uploader Tool.
- How to Guide for Language Test Project Managers.
- How to Guide for Language Testers.

Recommended browser

We recommend to use one of the following browsers:

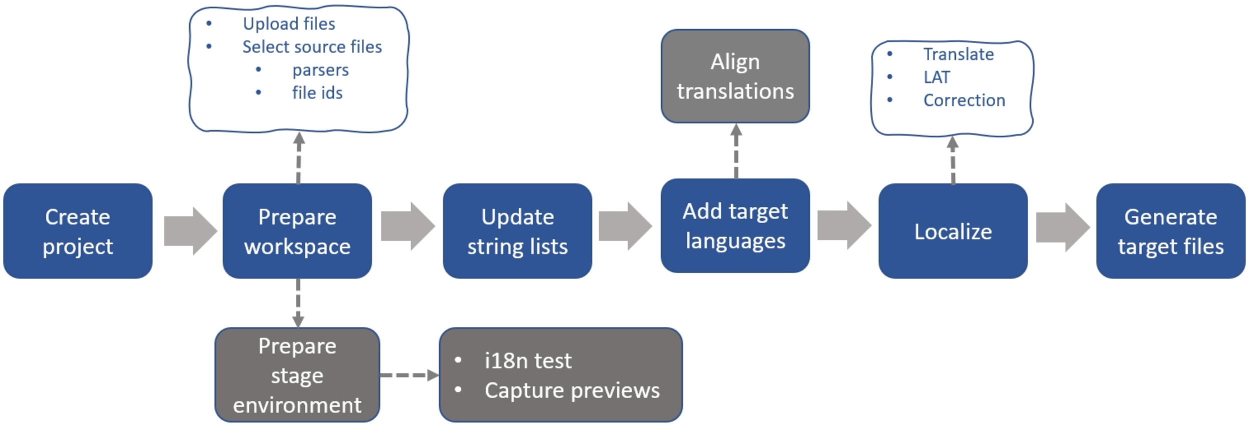
- Chrome
- Firefox

Internet Explorer is not supported.

Workflow

The workflow consists of the following steps

1. Create project
2. Prepare workspace
3. Prepare stage environment
4. Update string lists
5. Add target languages
6. Localize
7. Generate target files

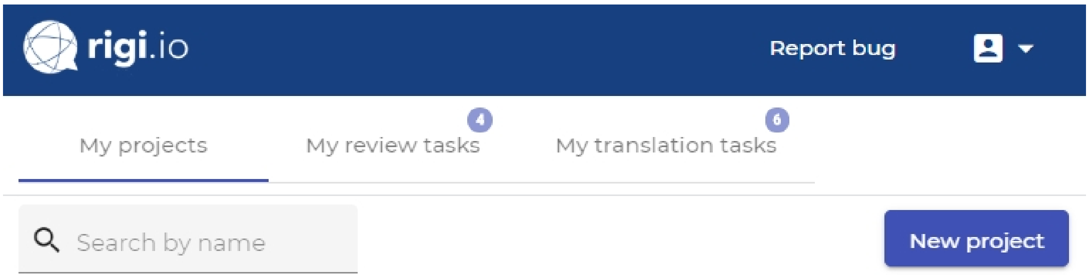


Continue with step-by-step instructions.

Steps - Create project

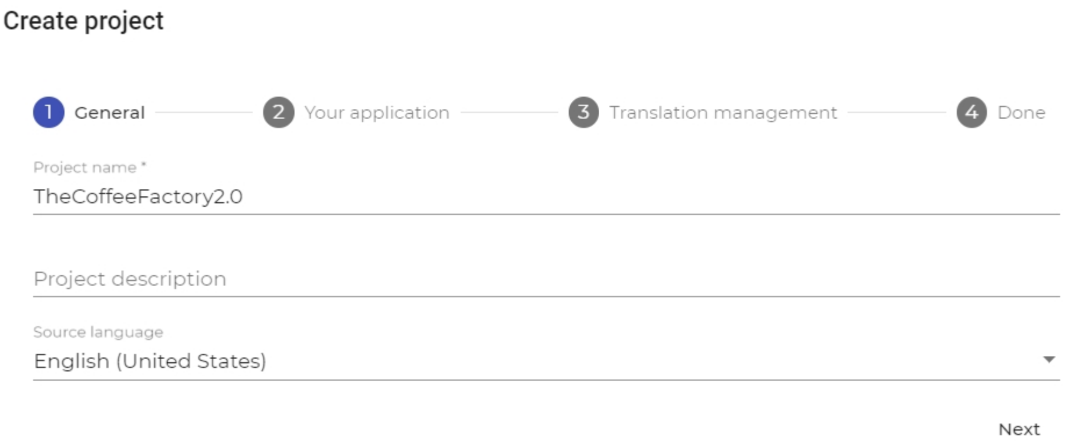
1. Login on Rigi server.

2. Select **Dashboard > My projects.**



3. Click **New project.**

    1. Enter the project name and source language.


Create project

1 General — 2 Your application — 3 Translation management — 4 Done

Project name *
TheCoffeeFactory2.0

Project description

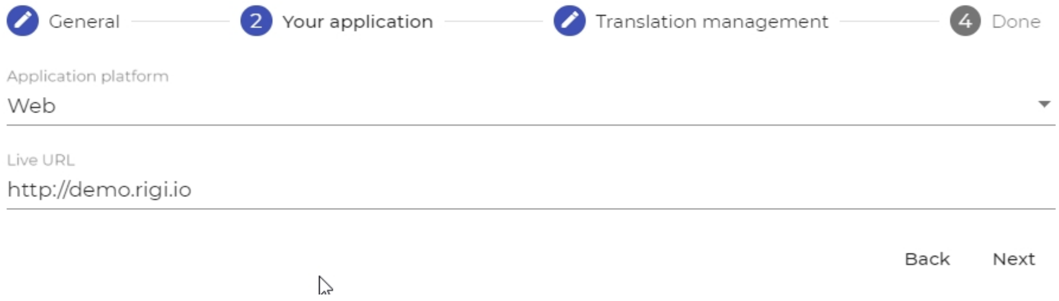
Source language
English (United States)

    2. Enter the platform.

    The selected platforms defines which characters will be used as Rigi tokens.

    In case of a web (or Silverlight) application you can enter the live URL to the staging application.

    If the live URL is not known at this moment, it is possible to enter it later via Settings > General.


General — 2 Your application — Translation management — 4 Done

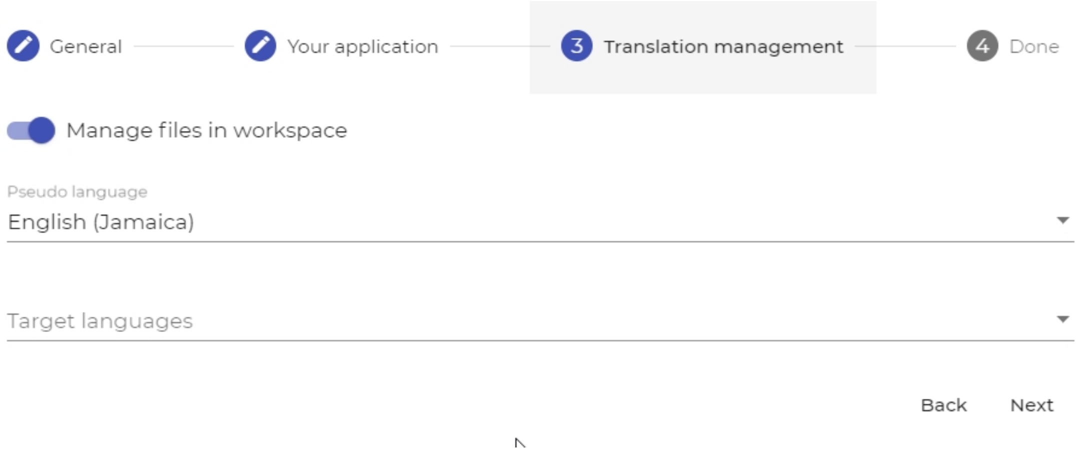
Application platform
Web

Live URL
http://demo.rigi.io

3. Specify the following

   - Manage files in workspace. *By default, the Rigi server parses the files. However, the file parsing may be managed outside the Rigi server in exceptional use cases (e.g. in SDL Passolo).*

   - Pseudo language. *The sole purpose of this language is to store Rigi tokens. It is only needed for localization purposes on a staging server.*

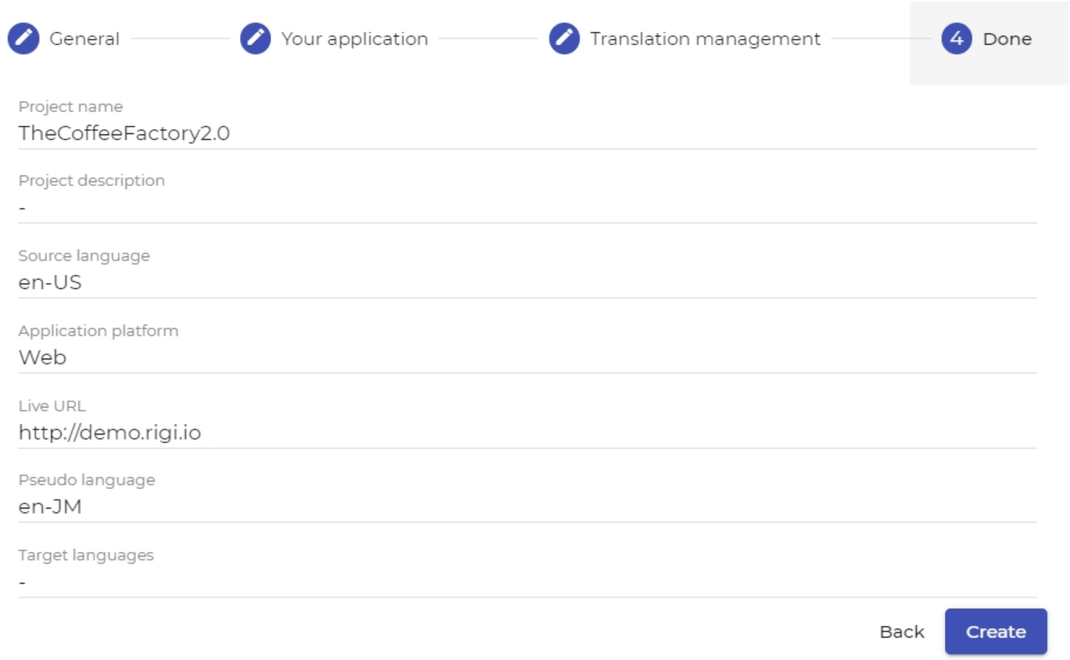


Manage files in workspace

Pseudo language
English (Jamaica)

Target languages

4. Check the overview



Project name
TheCoffeeFactory2.0

Project description
-

Source language
en-US

Application platform
Web

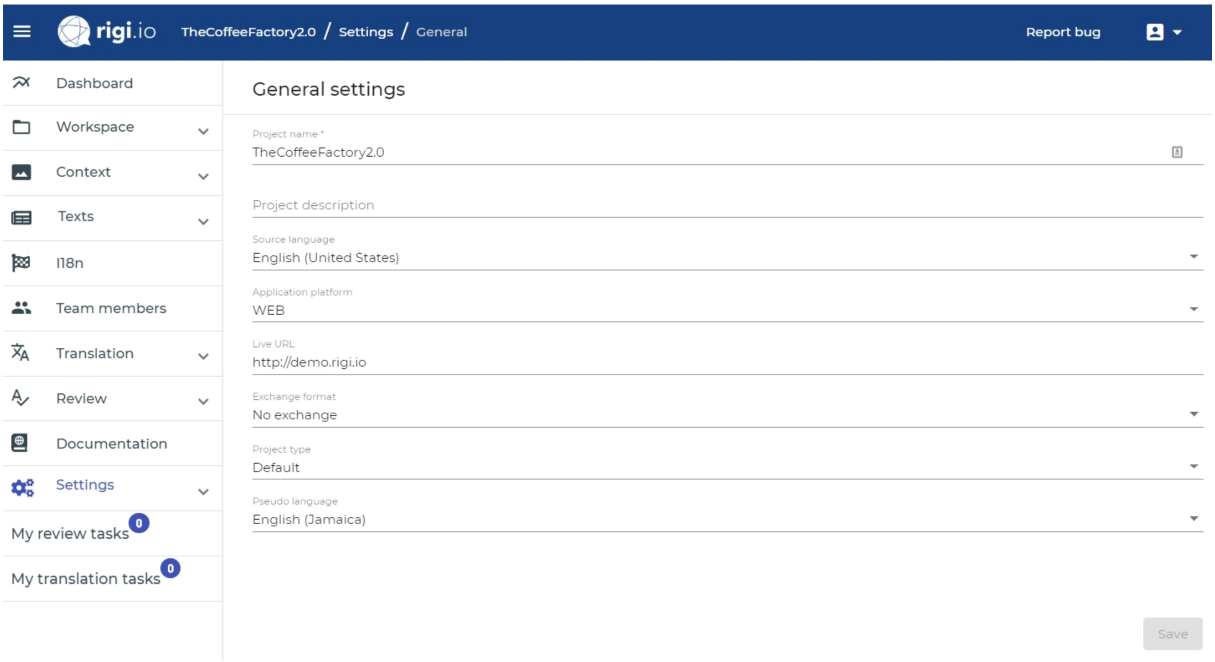
Live URL
http://demo.rigi.io

Pseudo language
en-JM

Target languages
-

4. Click **Create**.

   The project will be opened and showthe General settings. You could make changes here.



General settings

Project name *
TheCoffeeFactory2.0

Project description

Source language
English (United States)

Application platform
WEB

Live URL
http://demo.rigi.io

Exchange format
No exchange

Project type
Default

Pseudo language
English (Jamaica)

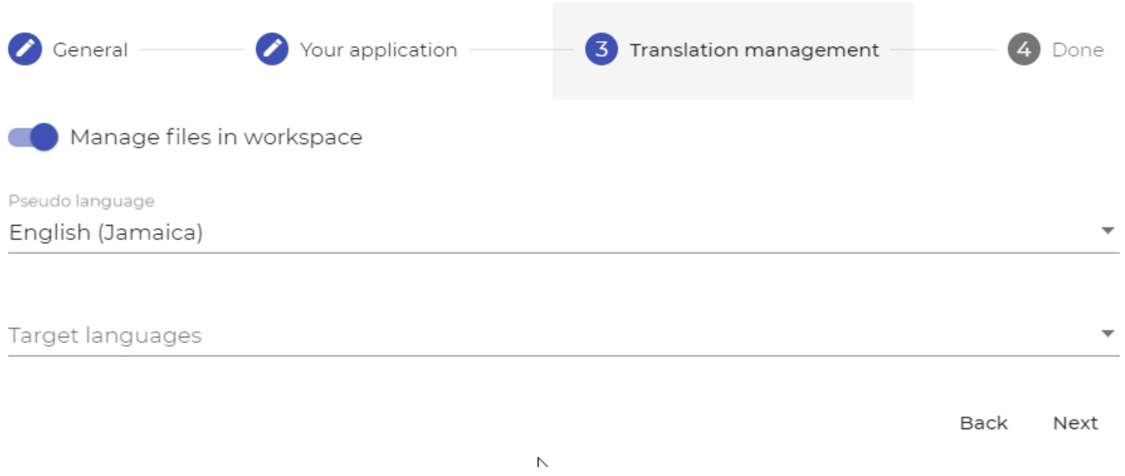
Continue with the steps to prepare the workspace.

Steps - Prepare workspace

Upload source and/or target resource files to the Rigi server. Select the source files and assign parsers and file IDs.

Prerequisites

- This step only applies to projects where the parsing is done on the Rigi server.



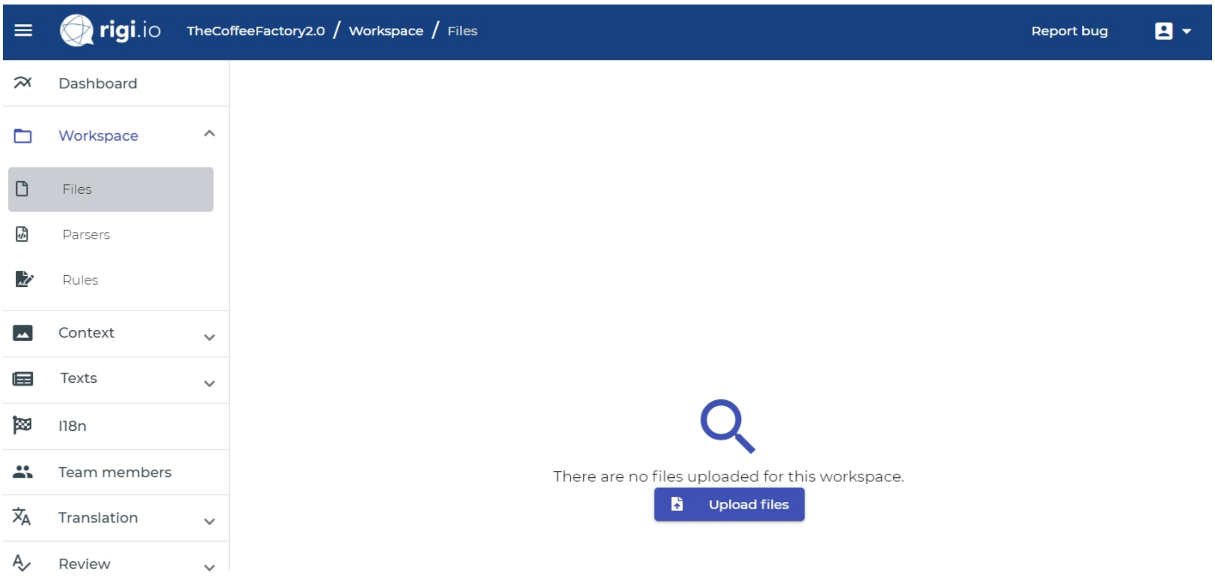
Manage files in workspace

Pseudo language: English (Jamaica)

Target languages

Steps

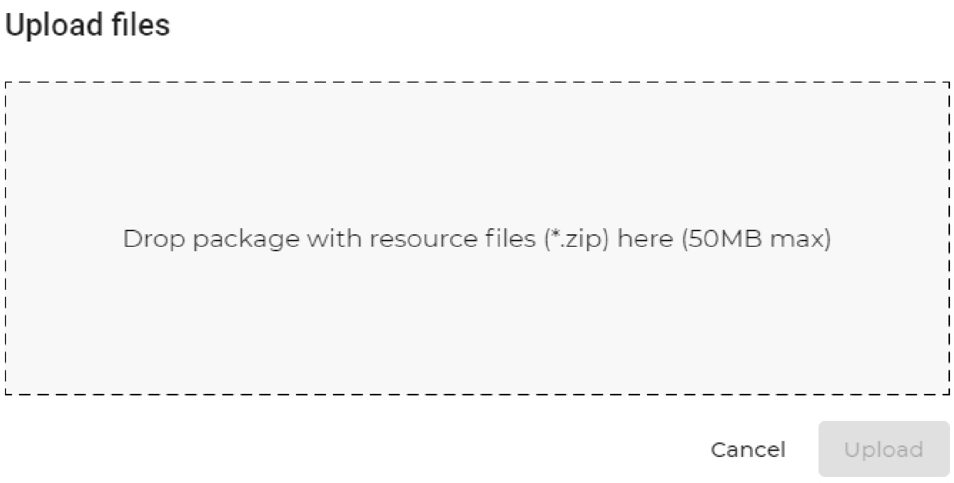
1. Open **Workspace > Files**.



There are no files uploaded for this workspace.

2. Click **Upload files** to upload source and/or target files.

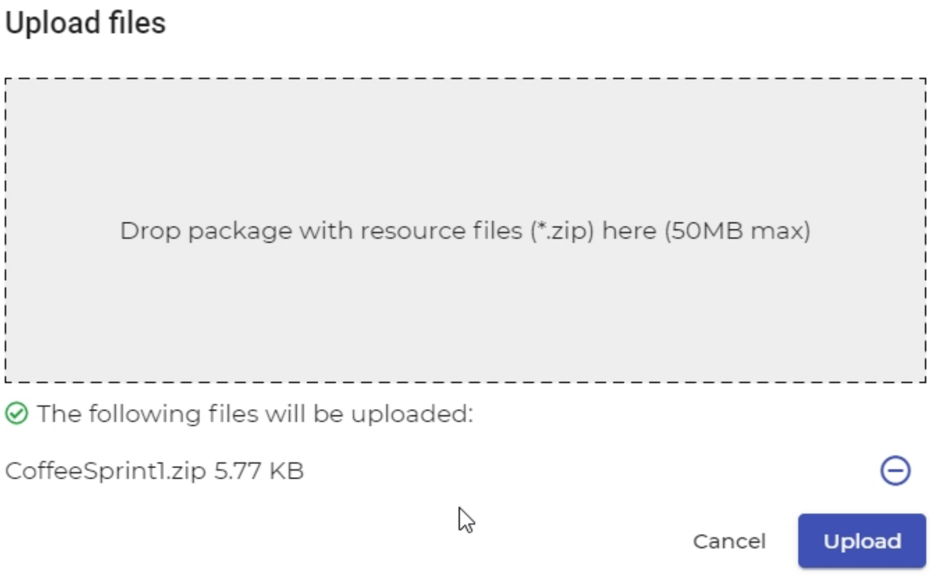
   Note: alternatively developers can upload files via the API.



Upload files

Drop package with resource files (*.zip) here (50MB max)

In this example we upload a ZIP file with the source (en.json) and one target language (de.json):



Upload files

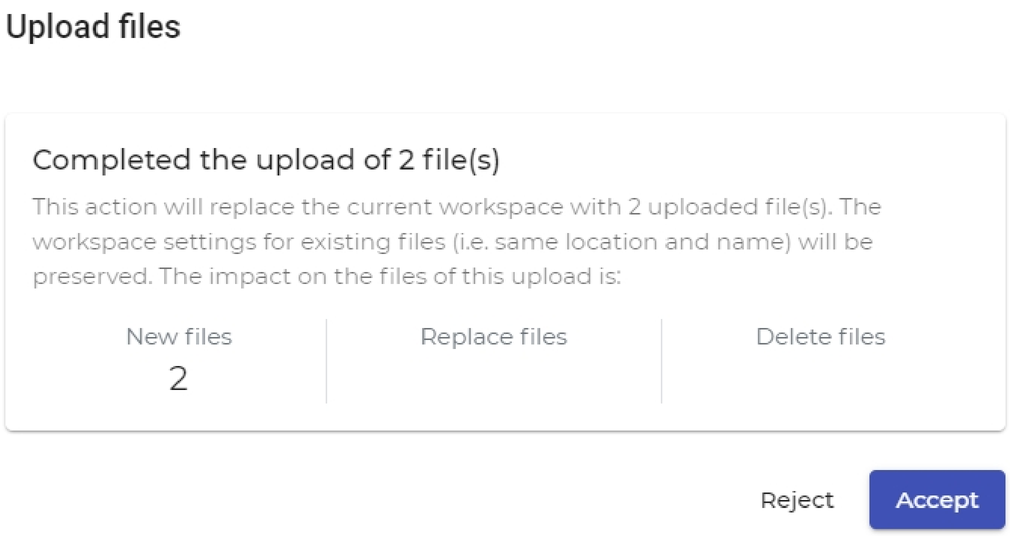
Drop package with resource files (*.zip) here (50MB max)

✓ The following files will be uploaded:

CoffeeSprint1.zip 5.77 KB

3. Click **Upload**.

   This will showa summary



Upload files

Completed the upload of 2 file(s)

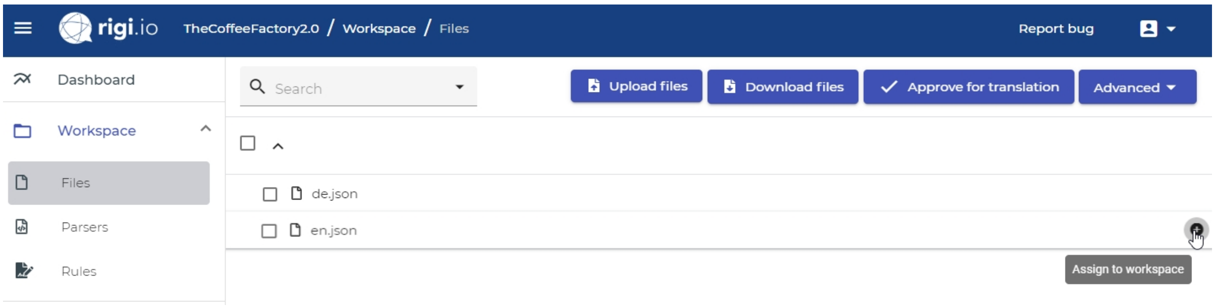
This action will replace the current workspace with 2 uploaded file(s). The workspace settings for existing files (i.e. same location and name) will be preserved. The impact on the files of this upload is:

New files	Replace files	Delete files
2		

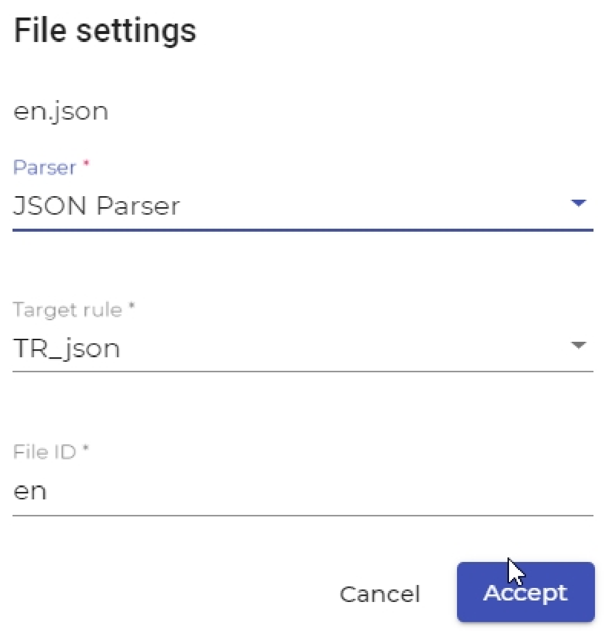
- *New files: a file with this full file path is not available in the current workspace (this is the case for new files).*
- *Replace files: this file was already available for this workspace. It will be replaced, but the parser, target rule and file ID remain the same. This is usual the case when new versions of files are uploaded.*
- *Delete files: some files in the workspace were missing in the uploaded ZIP. They will be removed and unassigned from the workspace.*

4. Click **Accept**.

5. Right click **Assign to workspace** to assign the selected source files to the workspace.



This will show the following dialog:



File settings

en.json

Parser *
JSON Parser

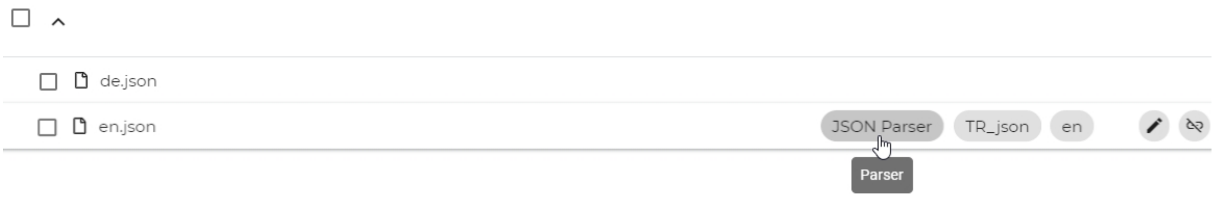
Target rule *
TR_json

File ID *
en

Enter the details for the added file:

1. Select parser. The current parser is pre-selected based on the file extension.
2. Select the target rule.
3. Enter the File ID. This identifier is used to construct the Rigi file identifier.

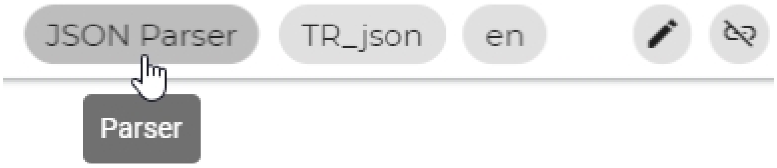
6. Click **Accept**.



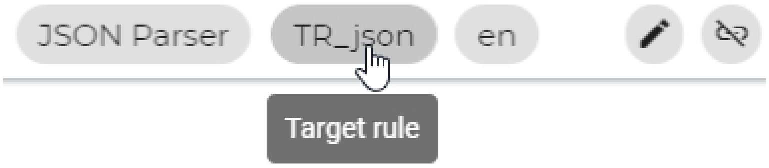
□ de.json
□ en.json

7. **Fine tune** the settings

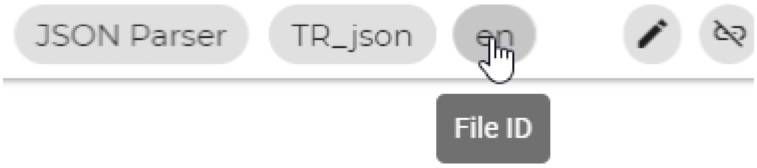
   1. Click **the parser badge** to verify/edit the parser settings.



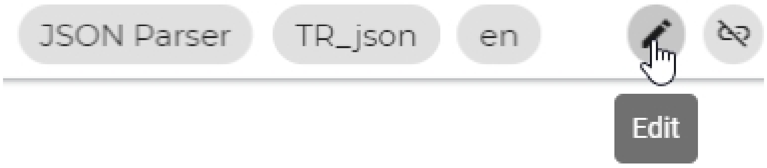
   2. Click **the target rule badge** to verify/edit the target rule.



   3. Click the **file ID badge**



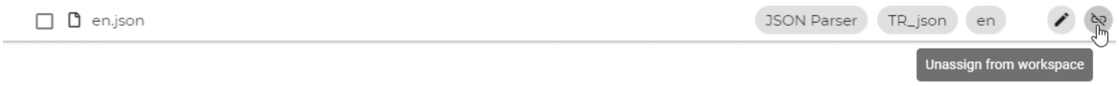
or **the edit icon**



to select the parser, target rule and/or specify the file ID.

Tip: use the export/import feature to mass rename file IDs.

4. Click **unassign from workspace** to unselect a file that shall not be parsed



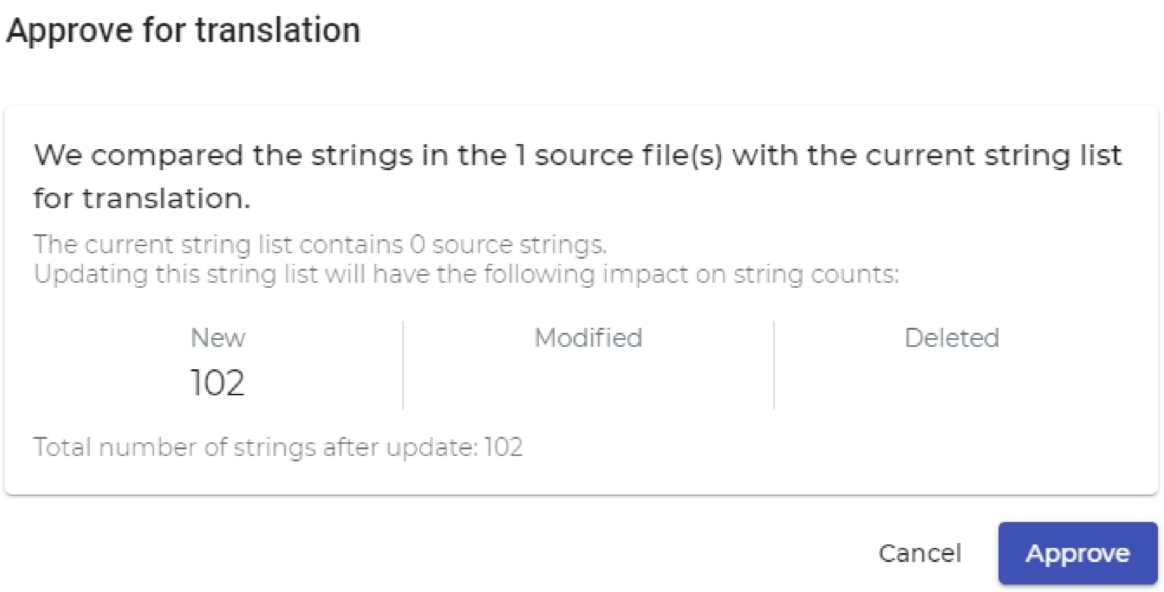
All source files are nowselected and have the correct parsers assigned.

8. Open **Workspace > Files**.

   Goal is to update the string list with texts from the selected source files in the workspace.

9. Click **Approve for translation**.

The following dialog appears:



Approve for translation

We compared the strings in the 1 source file(s) with the current string list for translation.

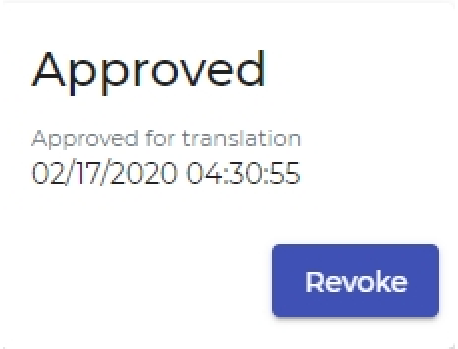
The current string list contains 0 source strings.
Updating this string list will have the following impact on string counts:

New	Modified	Deleted
102		

Total number of strings after update: 102

10. Click **Approve**.

    A panel appears at the bottom of the main menu:



Approved

Approved for translation
02/17/2020 04:30:55

You could revoke to the situation before the approval.

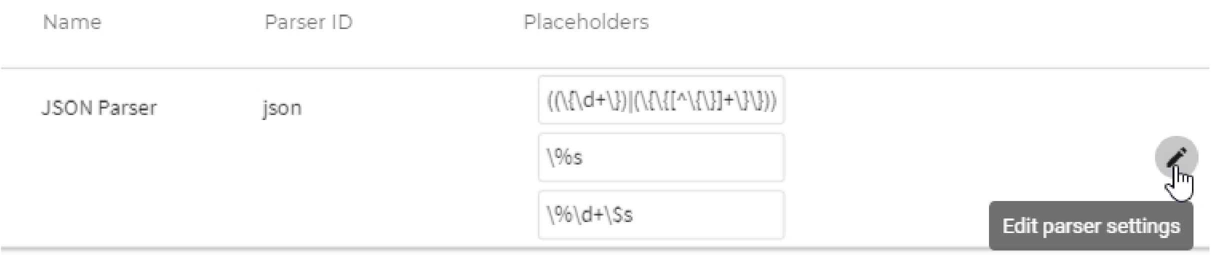
Continue with the steps to prepare the stage environment.
Parser settings

Rigi supports many file formats, e.g. json, properties, etc. and has a parser for each of those files.

Steps

1. Select **Workspace > Parsers**.

   This shows the parsers and settings that are used for the current project.

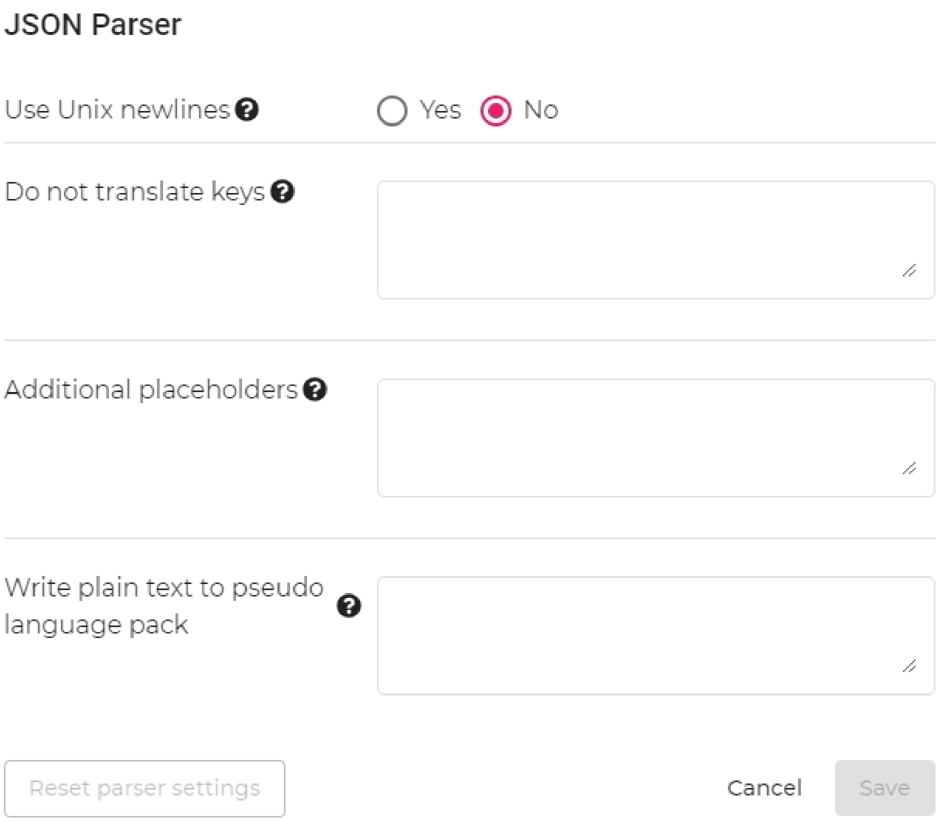


Name	Parser ID	Placeholders
JSON Parser	json	((\{\d+\})|(\{\{[^\{\}]+\}\})) \%s \%\d+\$s

The placeholders are regular expressions howvariables in UI texts are defined for this project. Each parser has default patterns defined. For example, in case of json, a variable could be defined as `{{name}}`*.*

2. Click **Edit parser settings**.

   This will open the parser settings dialog for the selected parser:



JSON Parser

Use Unix newlines ○ Yes ● No

Do not translate keys

Additional placeholders

Write plain text to pseudo language pack

   The ? - icons show help text for each parser specific setting.

3. Modify settings

4. Click **Save**.

Target rules

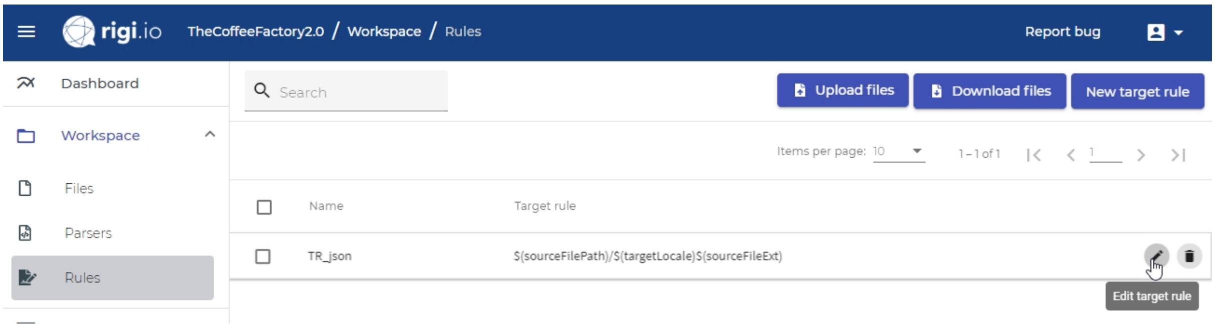
Target rules define the file path of the target files based on the file path of the source file.

In the following example, the source file is `.\en.json`.

Goal is to define a target rule, so that the target file is stored at the same location, where the filename is the language without locale. For example, in case of German (Germany), de-DE, the target file name shall be .\de`.json`.

Steps

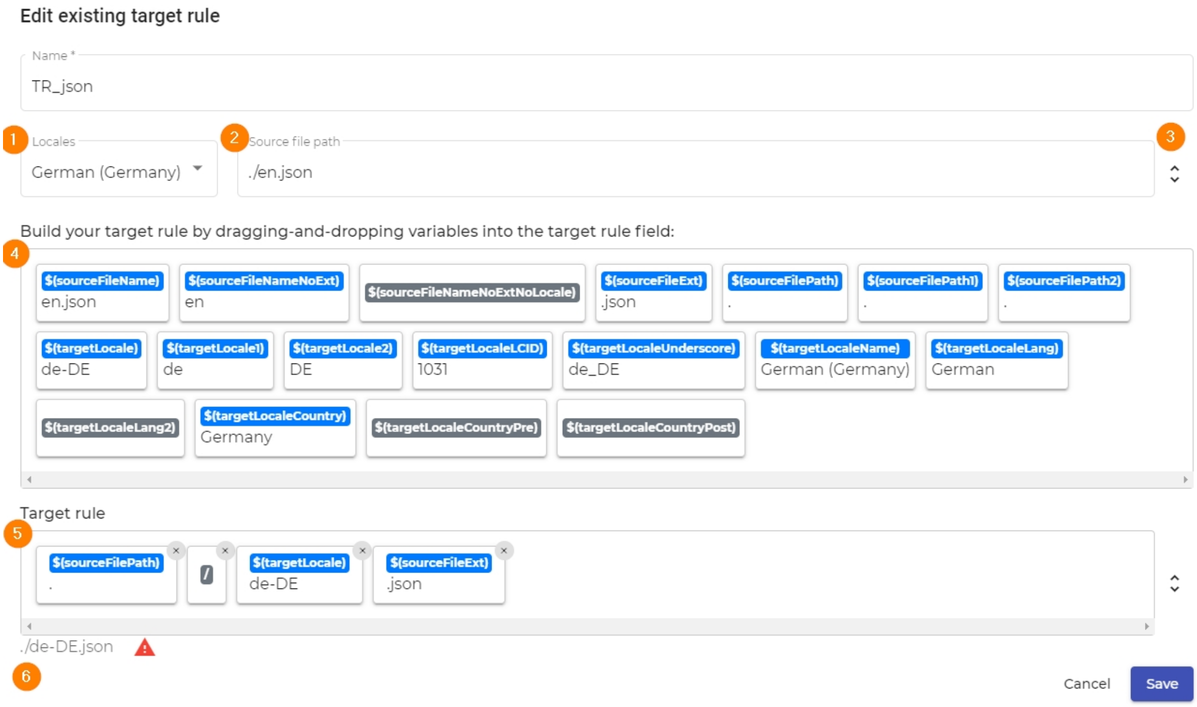
1. Select **Workspace > Rules**.



☐	Name	Target rule
☐	TR_json	${sourceFilePath}/${targetLocale}${sourceFileExt}

2. Click **Edit target rule**.

   This will open the rule editor:


Edit existing target rule

Name * TR_json

Locales: German (Germany)

Source file path: ./en.json

Build your target rule by dragging-and-dropping variables into the target rule field:

Variable	Value
${sourceFileName}	en.json
${sourceFileNameNoExt}	en
${sourceFileNameNoExtNoLocale}	
${sourceFileExt}	.json
${sourceFilePath}	.
${sourceFilePath1}	.
${sourceFilePath2}	.
${targetLocale}	de-DE
${targetLocale1}	de
${targetLocale2}	DE
${targetLocaleLCID}	1031
${targetLocaleUnderscore}	de_DE
${targetLocaleName}	German (Germany)
${targetLocaleLang}	German
${targetLocaleLang2}	
${targetLocaleCountry}	Germany
${targetLocaleCountryPre}	
${targetLocaleCountryPost}	

Target rule: ${sourceFilePath} . / ${targetLocale} de-DE ${sourceFileExt} .json

./de-DE.json ⚠

Items:

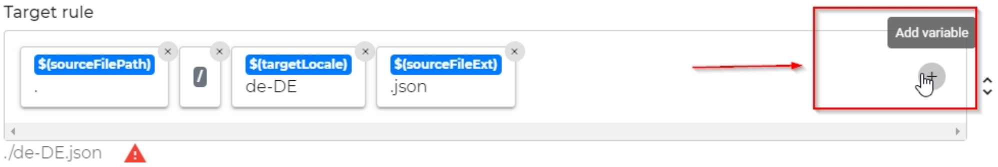
1. Select the locale to test the target rule with (6).
2. Enter the source file path to test the target rule with (6).
3. Instead of entering manually in (2), it is possible to select the file paths from the current workspace.
4. Variables and their current values based on (1) and (2). Variables that are

resolved to an empty string are marked as dark gray.

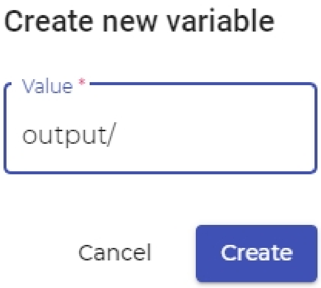
5. Build the target rule here.

    1. Drag and drop variables from (4), or

    2. Click **add variable**

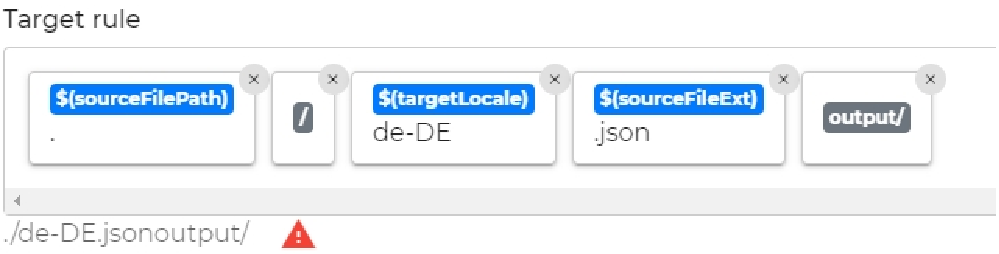


to **enter free text**

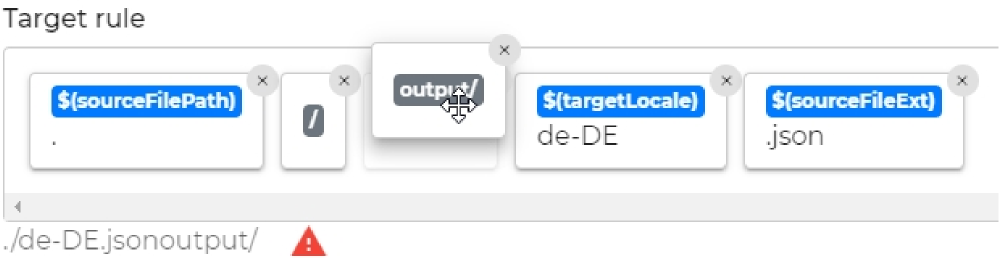


Value *: output/

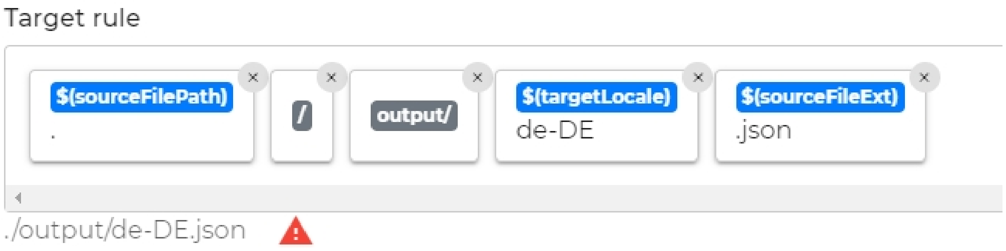
results in



and drag and drop to the right position:



like this:



Target rule

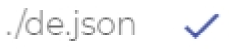
$(sourceFilePath) . | / | output/ | $(targetLocale) de-DE | $(sourceFileExt) .json

./output/de-DE.json ⚠

6. This is the resulting file path based on the variables in (1) and (2) amd the defined rule in (5).

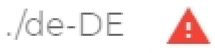
   There are icons shown behind the result:

   - **Check.** This means that the target file is available in the workspace. This is useful to verify if files are available in case you are planning an alignment.

     ./de.json ✓

   - **Warning.** This means that the target file is not available in the workspace.

     ./de-DE ⚠

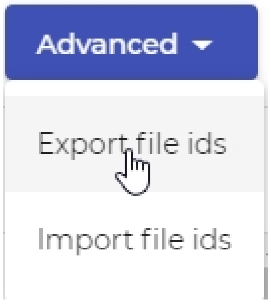
3. Click **Save**.

Mass File ID rename

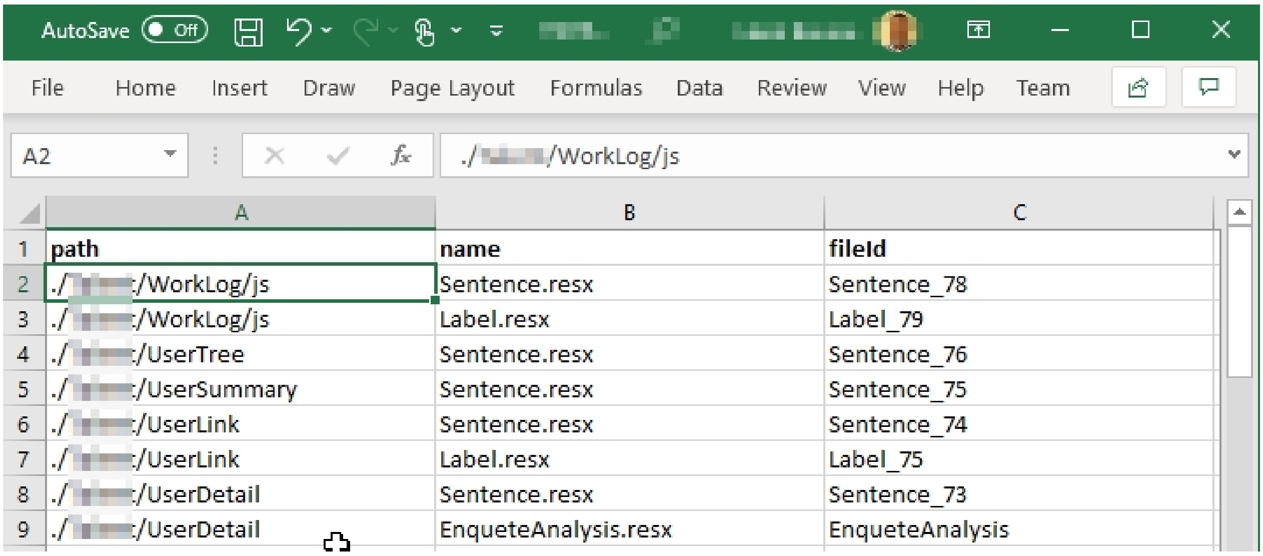
The export/import file IDs feature enables a mass rename of file IDs in an external editor (e.g. Excel).

Export

1. Click Workspace > Advanced > Export file ids



This exports the following file format:


AutoSave Off
File Home Insert Draw Page Layout Formulas Data Review View Help Team
A2
./[redacted]/WorkLog/js

	A	B	C
1	path	name	fileId
2	./[redacted]/WorkLog/js	Sentence.resx	Sentence_78
3	./[redacted]/WorkLog/js	Label.resx	Label_79
4	./[redacted]/UserTree	Sentence.resx	Sentence_76
5	./[redacted]/UserSummary	Sentence.resx	Sentence_75
6	./[redacted]/UserLink	Sentence.resx	Sentence_74
7	./[redacted]/UserLink	Label.resx	Label_75
8	./[redacted]/UserDetail	Sentence.resx	Sentence_73
9	./[redacted]/UserDetail	EnqueteAnalysis.resx	EnqueteAnalysis

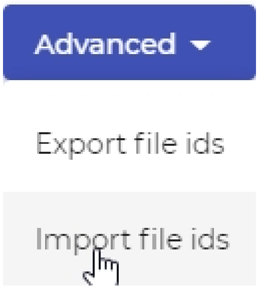
Edit

1. Open the exported file in Excel.

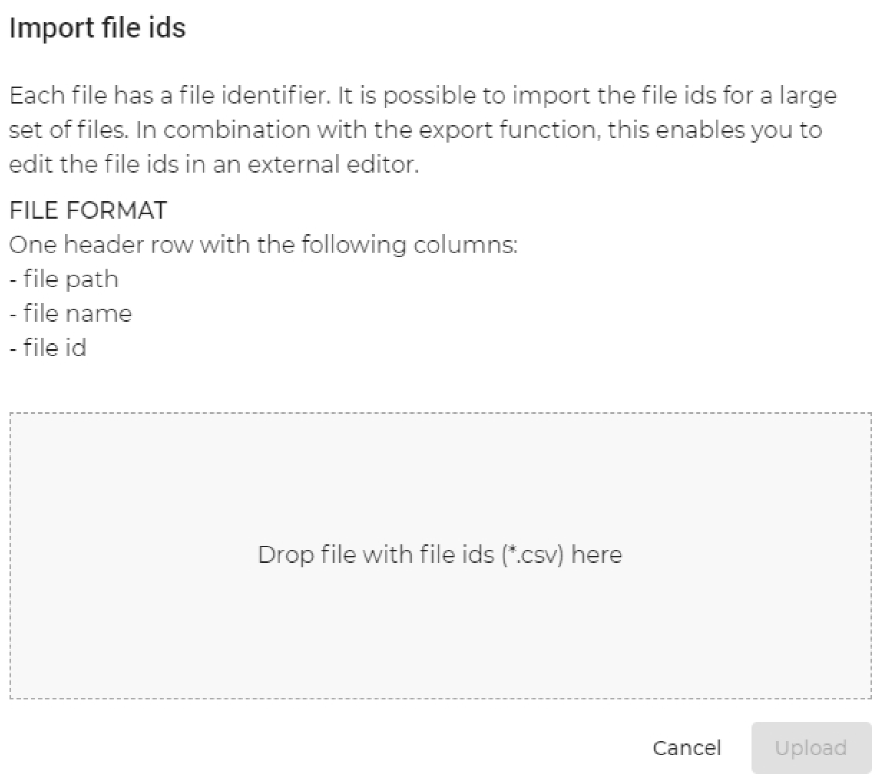
2. Change the file ids

Import

1. Click Workspace > Advanced > Import file ids



This opens the following dialog:



Import file ids

Each file has a file identifier. It is possible to import the file ids for a large set of files. In combination with the export function, this enables you to edit the file ids in an external editor.

FILE FORMAT
One header row with the following columns:
- file path
- file name
- file id

2. Select the file that you edited.
3. Click **Upload**

Nowthe file IDs in the project have been changed accordingly.

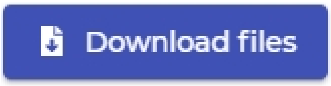
Steps - Prepare stage environment

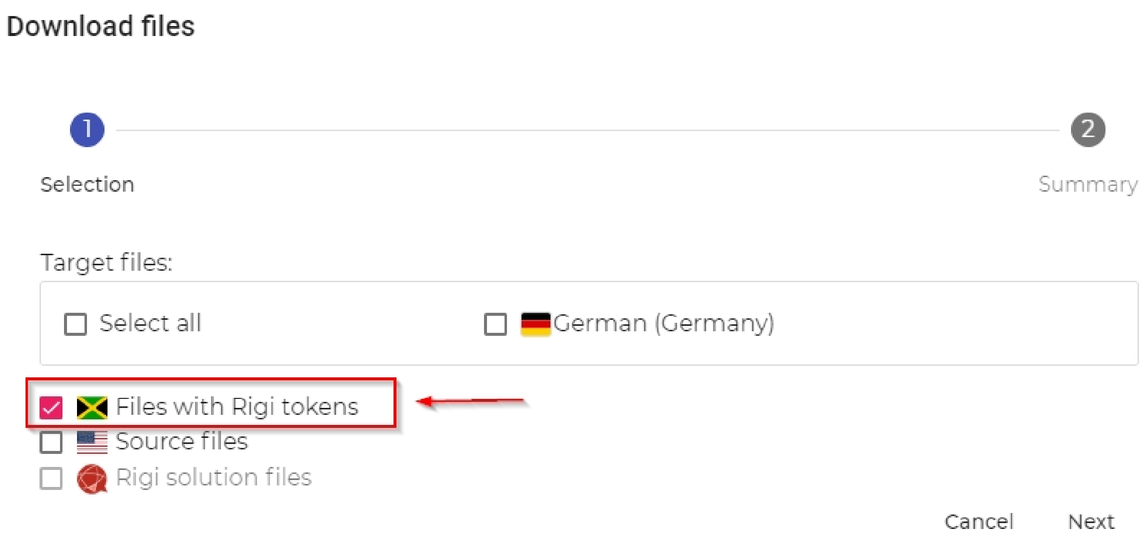
Prepare the staging server to enable the capturing of HTML Previews.

This requires that the Pseudo language can be loaded in the user interface. As a result, each captured HTML Preview contains Rigi tokens that allows Rigi to identify string identifiers.

1. Open **Workspace > Files**.

2. Click **Download files**.



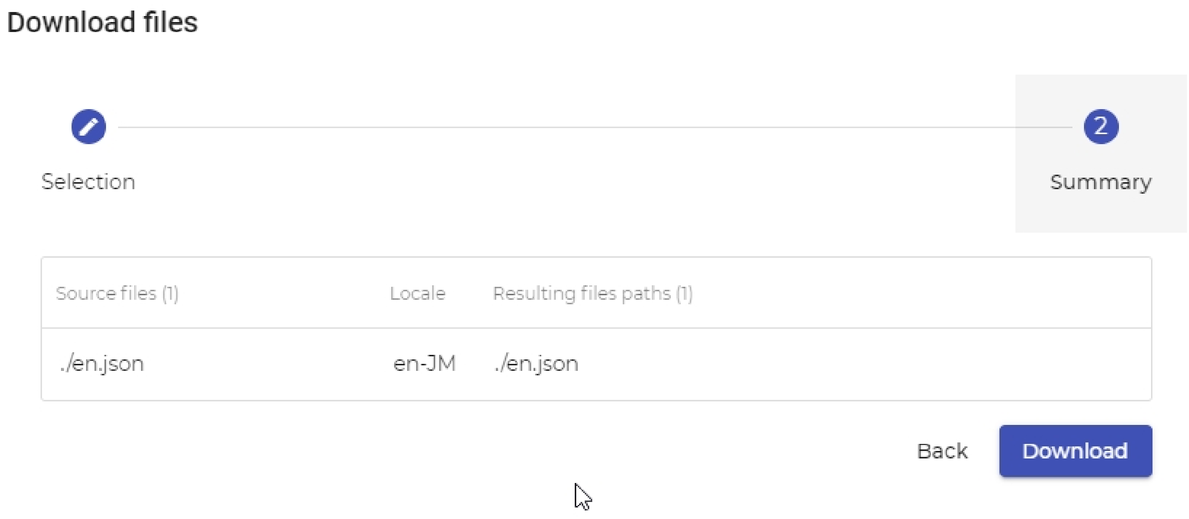


Target files:

□ Select all □ German (Germany)

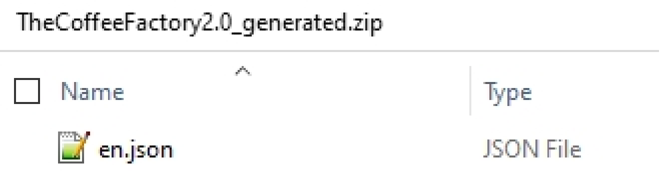
☑ Files with Rigi tokens
□ Source files
□ Rigi solution files

Click Next.



Source files (1)	Locale	Resulting files paths (1)
./en.json	en-JM	./en.json

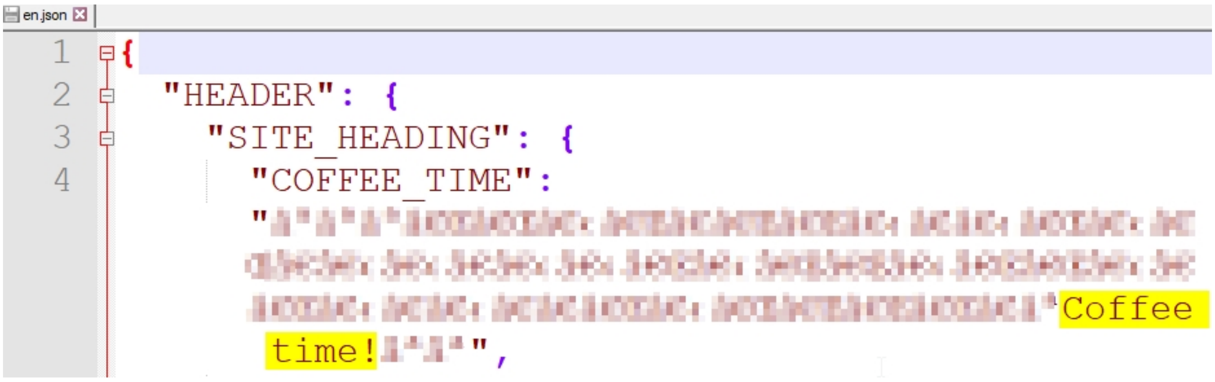
3. Click **Download**.



TheCoffeeFactory2.0_generated.zip

□ Name	Type
en.json	JSON File

This file contains Rigi tokens. Open the file in Notepad++ and change the encoding to ANSI. Notice that the Rigi tokens are there:



en.json

```
{
  "HEADER": {
    "SITE_HEADING": {
      "COFFEE_TIME":
      "[redacted]Coffee time![redacted]",
```

4. Deploy the Pseudo language to your staging server.

5. Verify the installation, see reference implementation below.

The Developers can now perform the following:

- Internationalization (i18n) test, see How to Guide to perform an internationalization (i18n) test.
- Capture HTML Previews, see How to Guide to capture HTML Previews.

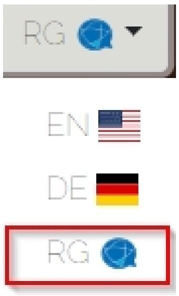
Continue with the steps to update the string lists.

Reference implementation

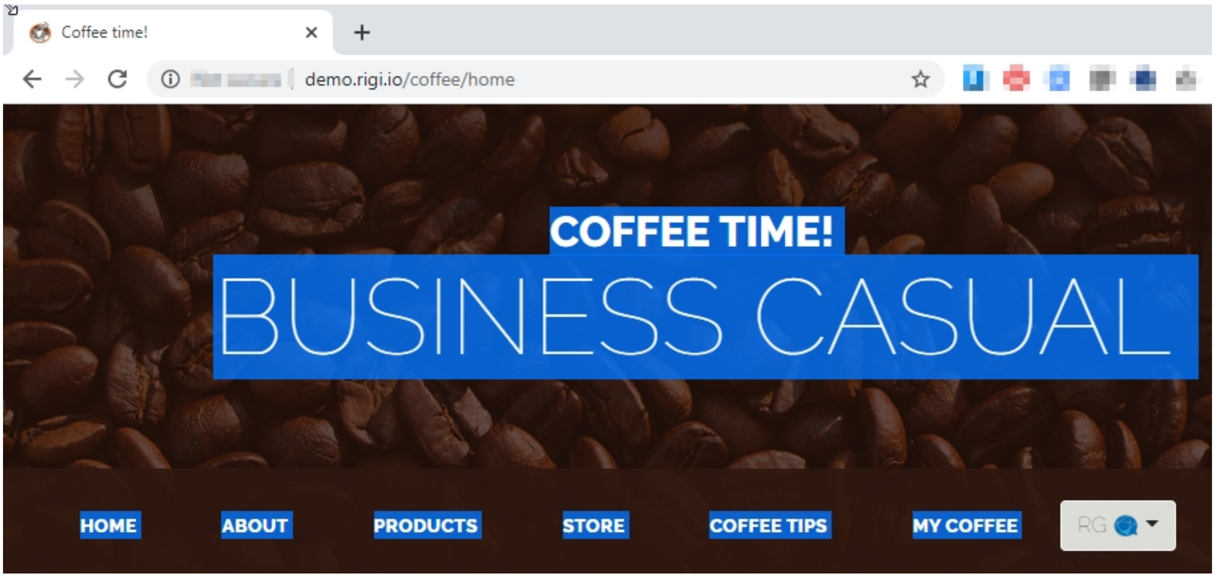
How to test if the language was deployed correctly?

1. Open http://demo.rigi.io

2. Select the Rigi langauge:



3. Select all texts (CTRL-A)

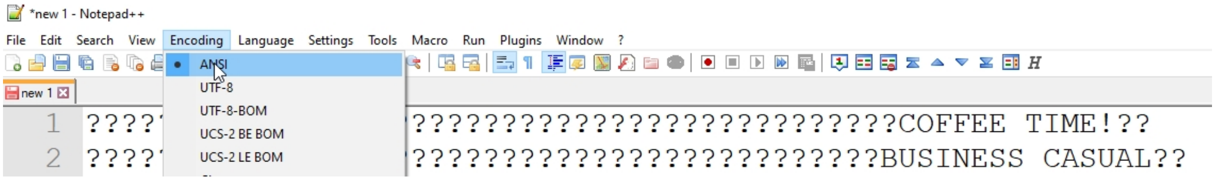


4. Open Notepad++.

5. Create a new document.

6. Select Encoding > ANSI

7. Paste the texts into Notepad++:



```
????                            ????????????????????????????COFFEE TIME!??
????                            ???????????????????????????BUSINESS CASUAL??
```

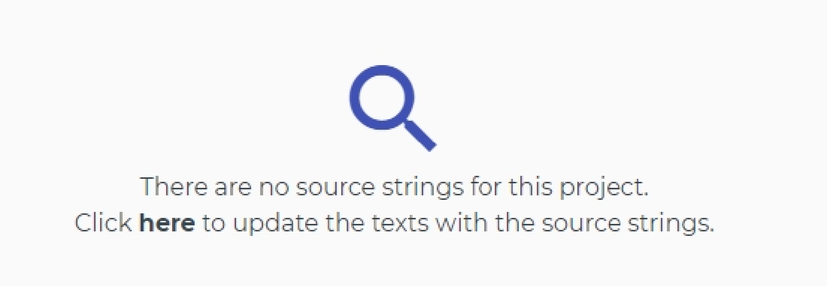
Steps - Update string lists

The workspace contains the correct source and target files. Each source file has a parser and file id.

Next step is to get the translations from the workspace into the translation management environment.

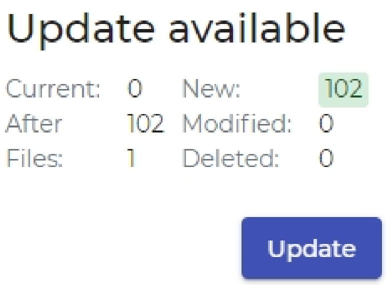
1. Open **Texts > Overview**.

   You see this message:



There are no source strings for this project.
Click **here** to update the texts with the source strings.

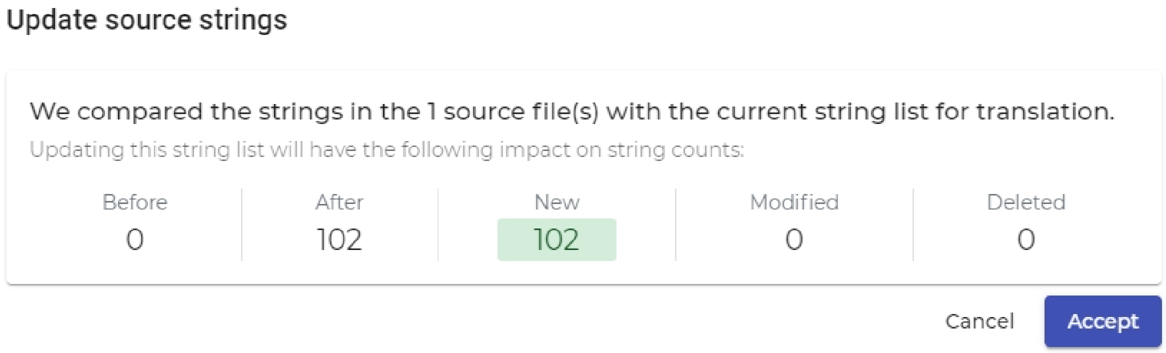
and a panel at the bottom of the main menu:



Update available

Current:	0	New:	102
After	102	Modified:	0
Files:	1	Deleted:	0

2. Click **Update**.



Update source strings

We compared the strings in the 1 source file(s) with the current string list for translation.
Updating this string list will have the following impact on string counts:

Before	After	New	Modified	Deleted
0	102	102	0	0

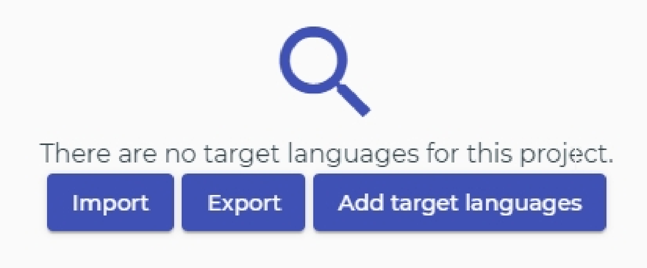
3. **Verify if those changes match your expectations!**
   - Before: number of source strings in the string list before this operation.
   - After: number of source strings in the string list after this operation.
   - New: number of strings with string IDs that are new.
   - Modified: number of source strings with an existing string ID, but different source text.

- Deleted: Number of source strings in the list that will be deleted by this operation.

If these numbers do not look OK, please double check it with the developers!

4. Click **Accept**.

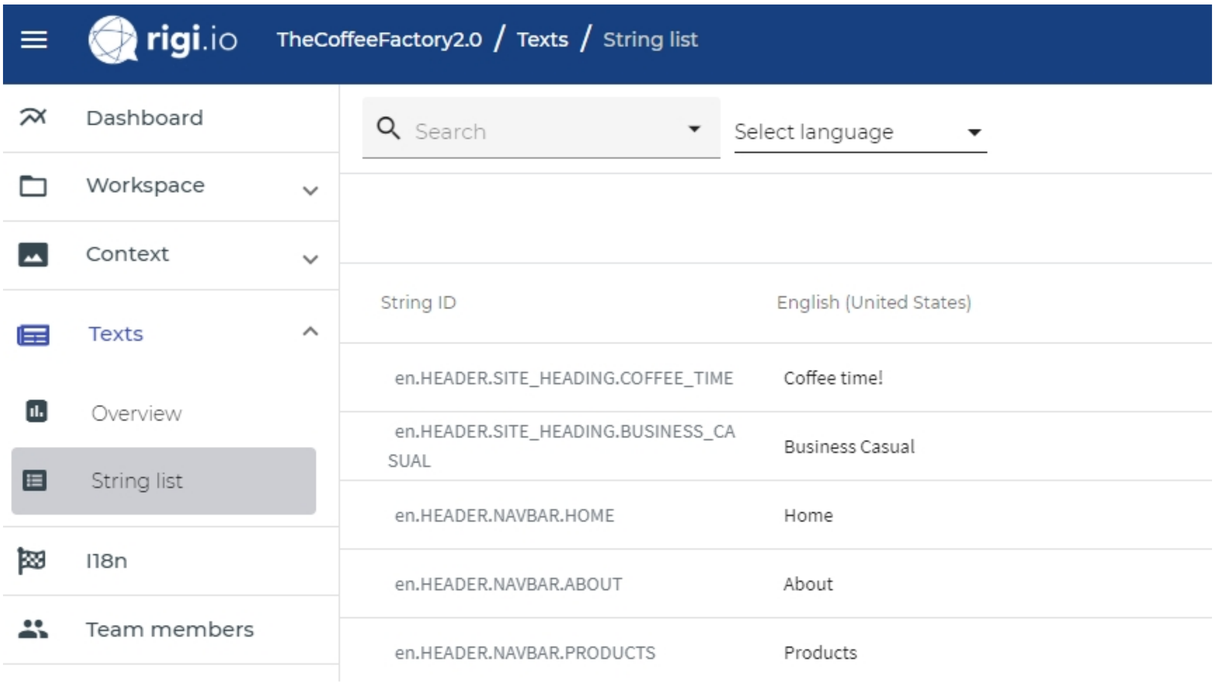
   You see this message:



There are no target languages for this project.

5. Open **Texts > String list**.

   You see the source strings:



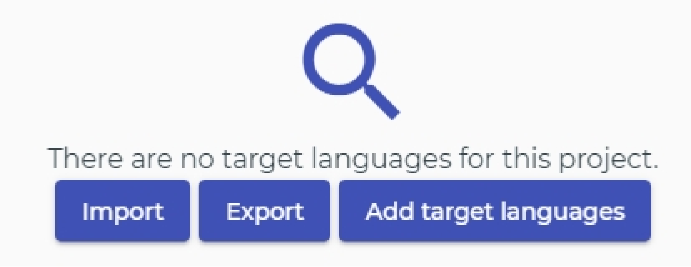
String ID	English (United States)
en.HEADER.SITE_HEADING.COFFEE_TIME	Coffee time!
en.HEADER.SITE_HEADING.BUSINESS_CASUAL	Business Casual
en.HEADER.NAVBAR.HOME	Home
en.HEADER.NAVBAR.ABOUT	About
en.HEADER.NAVBAR.PRODUCTS	Products

6. The Project Manager arranges that the following steps are performed in succession:
   1. Translation,
   2. Language Acceptance Test,
   3. Correction of reported issues.

Continue with the steps to add the target languages.

Steps - Add target languages

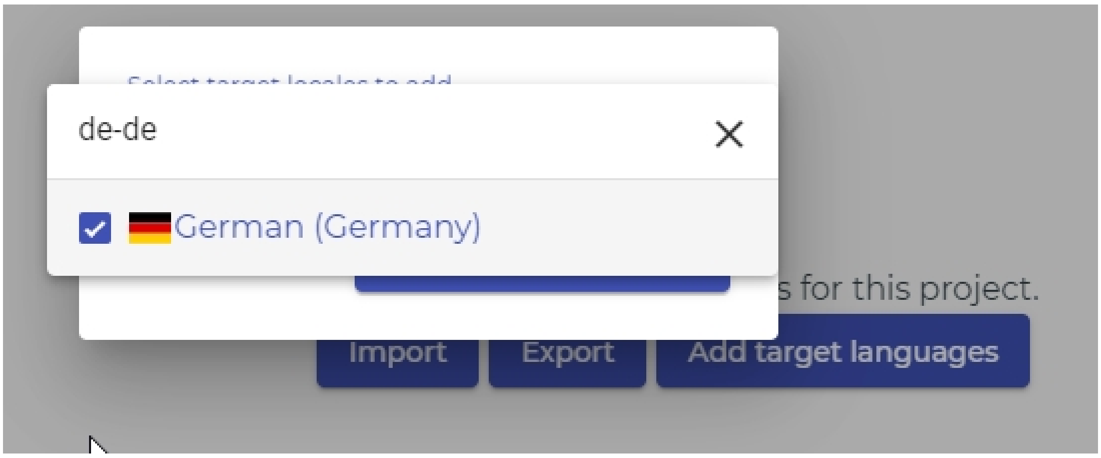
1. Open **Texts > Overview**.

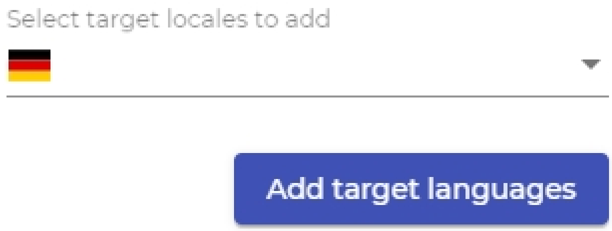


There are no target languages for this project.

2. Click **Add target languages**.

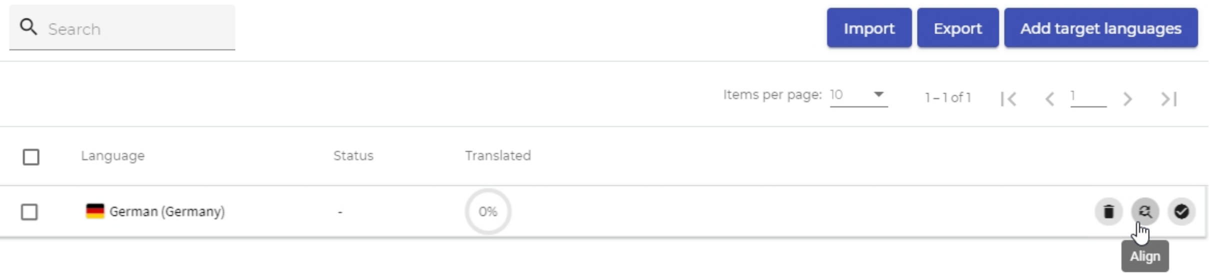
Tip! You can type the country code (e.g. de-DE, or nl-NL) to search for languages.





3. Click **Add target languages**.

The language is added to the string list, but it does not have translations yet.



□	Language	Status	Translated
□	German (Germany)	-	0%

4. Optional: **align** the translations with existing translations, see Align with translations.

Continue with the steps to localize.

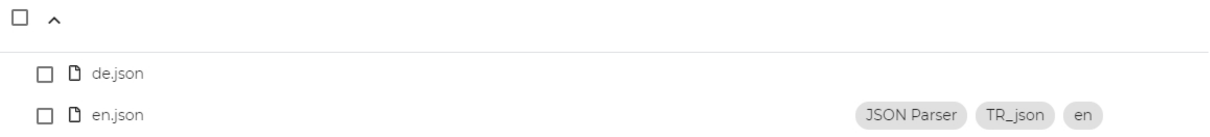
Align with translations

Alignment populates the string list for a particular language with translations from existing files in the workspace. The matching is ID-based and therefore exact.

This operation is usually done once when a new language is added to the project workspace (with existing translations).

1. Open **Workspace > Files**.

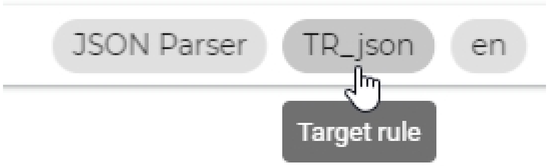
   In the example belowthe source file is `en.json` *and there is one file with German translations (*`de.json`*).*



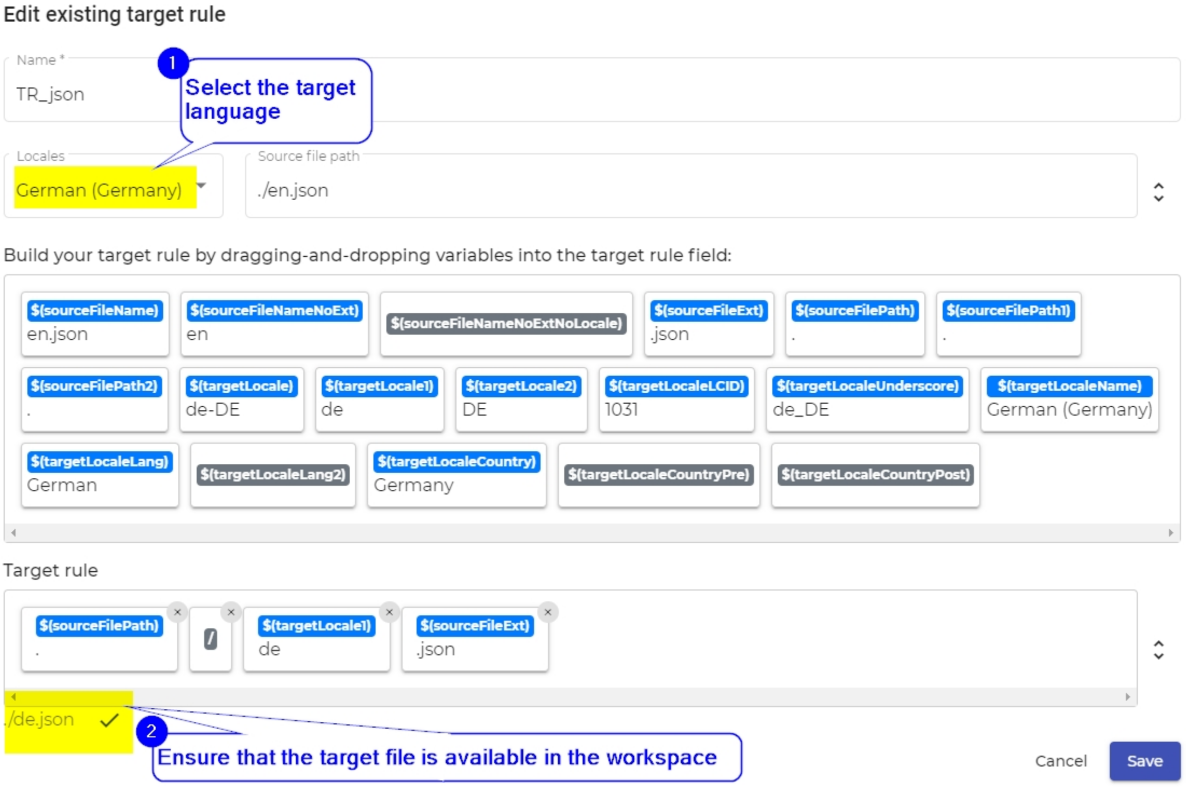
□ de.json

□ en.json JSON Parser TR_json en

2. Open the target rule



3. Verify that the target rule is correct



Edit existing target rule

Name *: TR_json

1 Select the target language

Locales: German (Germany)

Source file path: ./en.json

Build your target rule by dragging-and-dropping variables into the target rule field:

${sourceFileName}	en.json
${sourceFileNameNoExt}	en
${sourceFileNameNoExtNoLocale}	
${sourceFileExt}	.json
${sourceFilePath}	.
${sourceFilePath1}	.
${sourceFilePath2}	.
${targetLocale}	de-DE
${targetLocale1}	de
${targetLocale2}	DE
${targetLocaleLCID}	1031
${targetLocaleUnderscore}	de_DE
${targetLocaleName}	German (Germany)
${targetLocaleLang}	German
${targetLocaleLang2}	
${targetLocaleCountry}	Germany
${targetLocaleCountryPre}	
${targetLocaleCountryPost}	

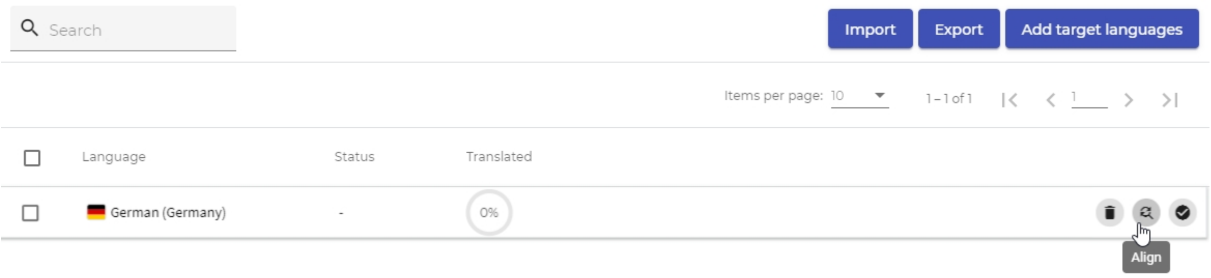
Target rule

${sourceFilePath} . / ${targetLocale1} de ${sourceFileExt} .json

./de.json ✓

2 Ensure that the target file is available in the workspace

4. **Open Texts > Overview**

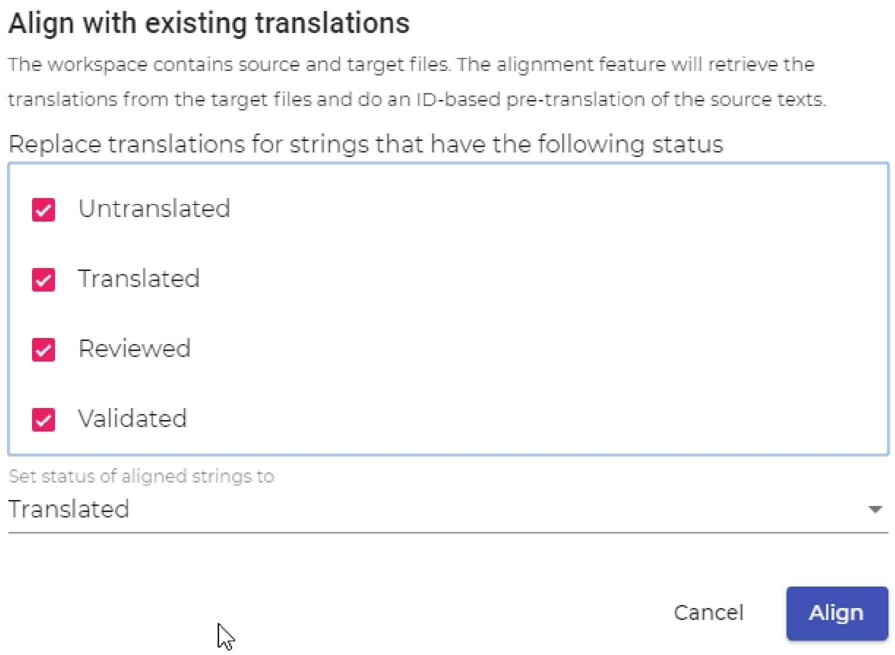


Items per page: 10 1 – 1 of 1

	Language	Status	Translated
☐	German (Germany)	-	0%

5. Click **Align**.

You have the option to ignore certain strings to be aligned.



Align with existing translations

The workspace contains source and target files. The alignment feature will retrieve the translations from the target files and do an ID-based pre-translation of the source texts.

Replace translations for strings that have the following status

☑ Untranslated

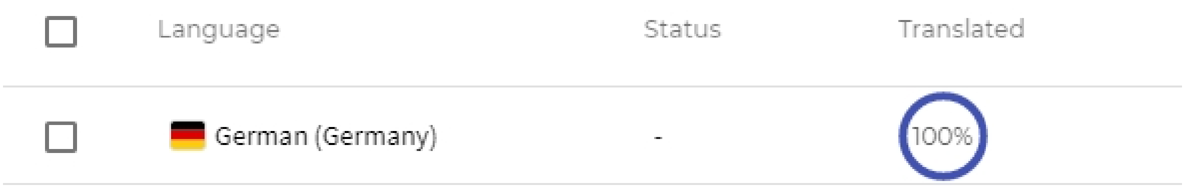
☑ Translated

☑ Reviewed

☑ Validated

Set status of aligned strings to: Translated

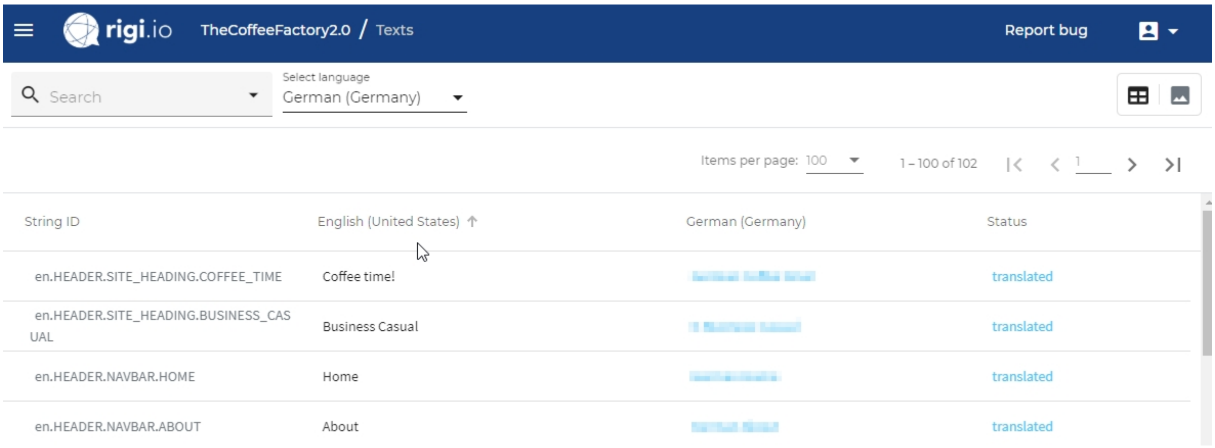
6. Click **Align**.



□	Language	Status	Translated
□	German (Germany)	-	100%

7. Click the language to open the string list

The translations are there:



Select language: German (Germany)

Items per page: 100   1 – 100 of 102   1

String ID	English (United States)	German (Germany)	Status
en.HEADER.SITE_HEADING.COFFEE_TIME	Coffee time!	[redacted]	translated
en.HEADER.SITE_HEADING.BUSINESS_CASUAL	Business Casual	[redacted]	translated
en.HEADER.NAVBAR.HOME	Home	[redacted]	translated
en.HEADER.NAVBAR.ABOUT	About	[redacted]	translated

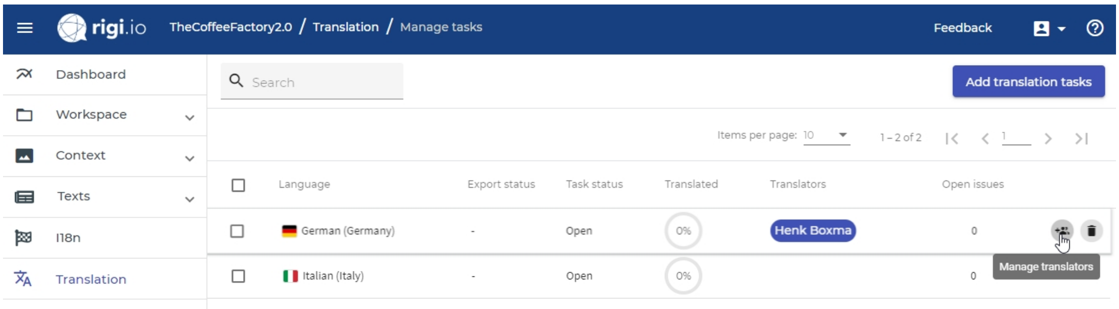
Steps - Localize

The Project Manager orchestrates that the following localization activities are performed in succession:

1. Translate.

Option 1: Rigi built-in editor:

1. Specify translators via Translation > Add translation tasks



□	Language	Export status	Task status	Translated	Translators	Open issues
□	German (Germany)	-	Open	0%	Henk Boxma	0
□	Italian (Italy)	-	Open	0%		0

*Translators get an email and see the translation tasks in **Dashboard > My translation tasks**.*

2. Translators click the link to their task.

   This opens the built-in Rigi editor with the Rigi Viewer where the translator can translate in-context.

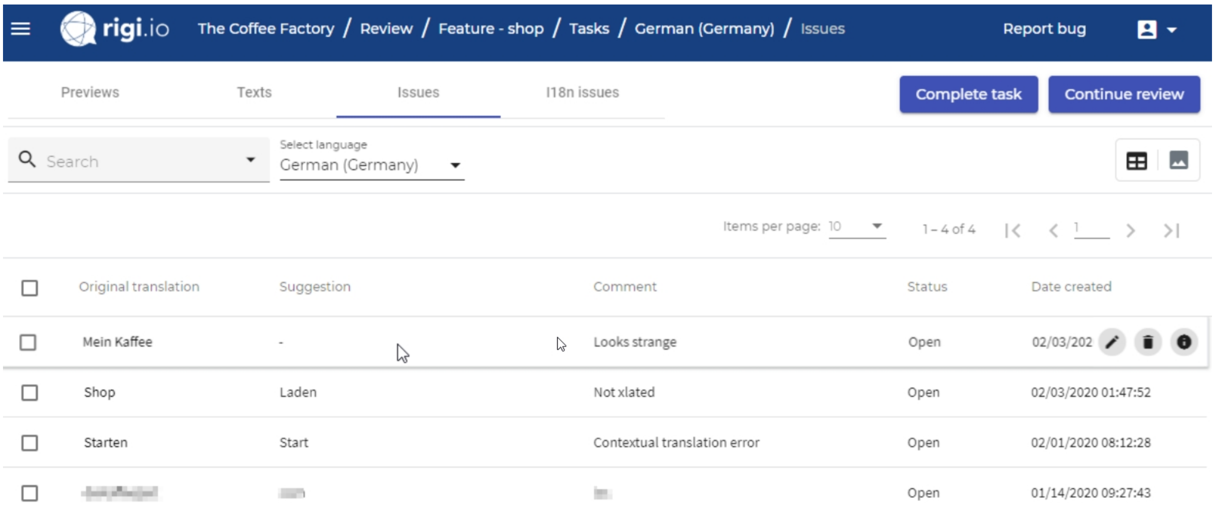
Option 2: connectors to third party translation editors

Vendor	Editor	Strategy	Process
rigi.io	Built-in Rigi editor	Translate directly on the Rigi server.	See above.
memoq	memoQ	• Export/Import XML 1.2	• How to Guide for Project Managers (TBD). • How to Guide for Translators (TBD).
SDL	SDL Passolo	• Parse in Passolo • Parse on Rigi server.	• How to Guide for Project Managers. • How to Guide for Translators.
	SDL Trados Studio	• Export/Import	• How to Guide for

		XLIFF	Project Managers (TBD) - [How to Guide for Translators]().
translate5	Translate5	- Export/Import XML 1.2	- How to Guide for Project Managers (TBD). - How to Guide for Translators (TBD).
XTM International	XTM	- Export/Import XLIFF - Export/Import XML 1.1	- How to Guide for Project Managers (TBD). - How to Guide for Translators (TBD).

2. Language Acceptance Test. See [How Guide for Language Test Project Managers]().

   This results in issues that are reported by the reviewers.

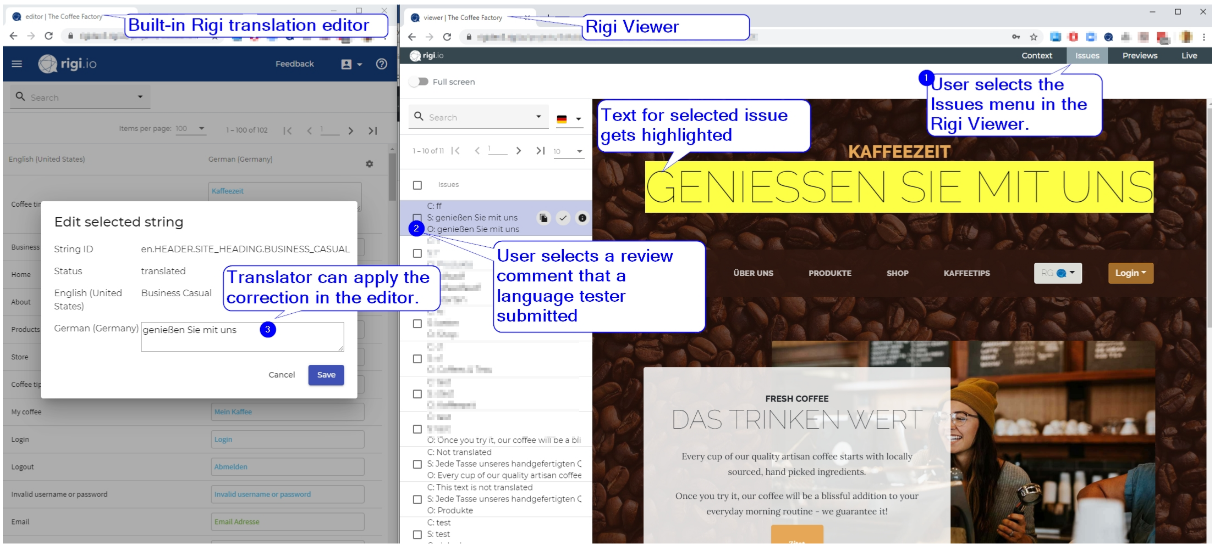


☐	Original translation	Suggestion	Comment	Status	Date created
☐	Mein Kaffee	-	Looks strange	Open	02/03/202
☐	Shop	Laden	Not xlated	Open	02/03/2020 01:47:52
☐	Starten	Start	Contextual translation error	Open	02/01/2020 08:12:28
☐	[redacted]	[redacted]	[redacted]	Open	01/14/2020 09:27:43

3. Correction.

   Translators can viewthe reported issues in the Rigi Viewer that is connected to their translation environment.

   The following screenshot shows the Rigi Viewer that is connected to the built-in Rigi Editor.



Built-in Rigi translation editor

Rigi Viewer

1 User selects the Issues menu in the Rigi Viewer.

Text for selected issue gets highlighted

2 User selects a review comment that a language tester submitted

3 Translator can apply the correction in the editor.

Edit selected string

String ID	en.HEADER.SITE_HEADING.BUSINESS_CASUAL
Status	translated
English (United States)	Business Casual
German (Germany)	genießen Sie mit uns

Cancel Save

C: ff
S: genießen Sie mit uns
O: genießen Sie mit uns

O: Once you try it, our coffee will be a bli

C: Not translated
S: Jede Tasse unseres handgefertigten C
O: Every cup of our quality artisan coffee

C: This text is not translated
S: Jede Tasse unseres handgefertigten C
O: Produkte

C: test
S: test

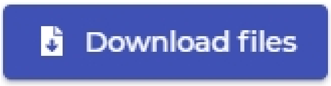
Continue with the steps to generate the target files.

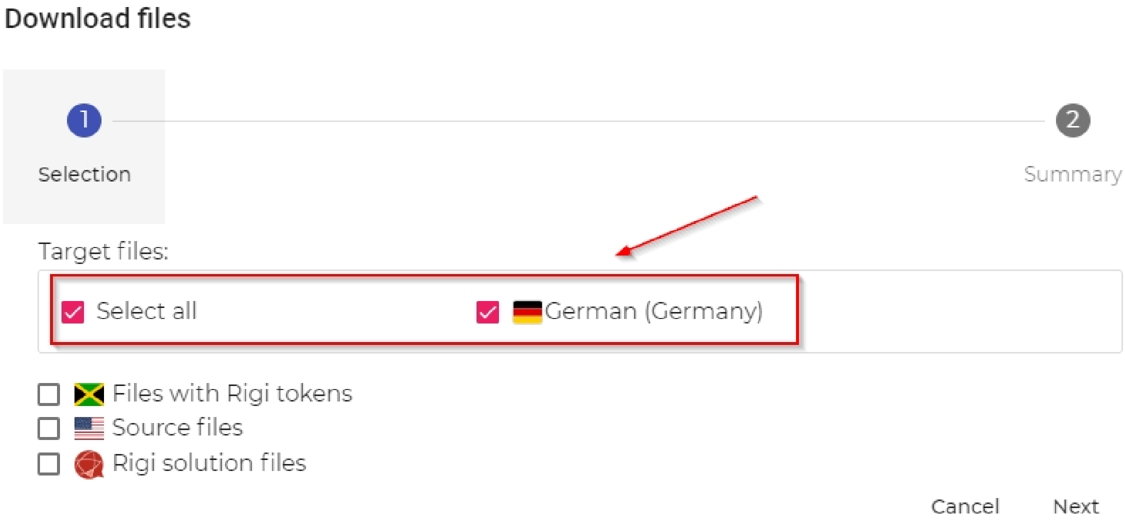
Steps - Generate target files

The translations are available in the database (Texts / string lists). This process step will generate resource files with translations.

1. Open **Workspace > Files**.

2. Click **Download files**.





Download files

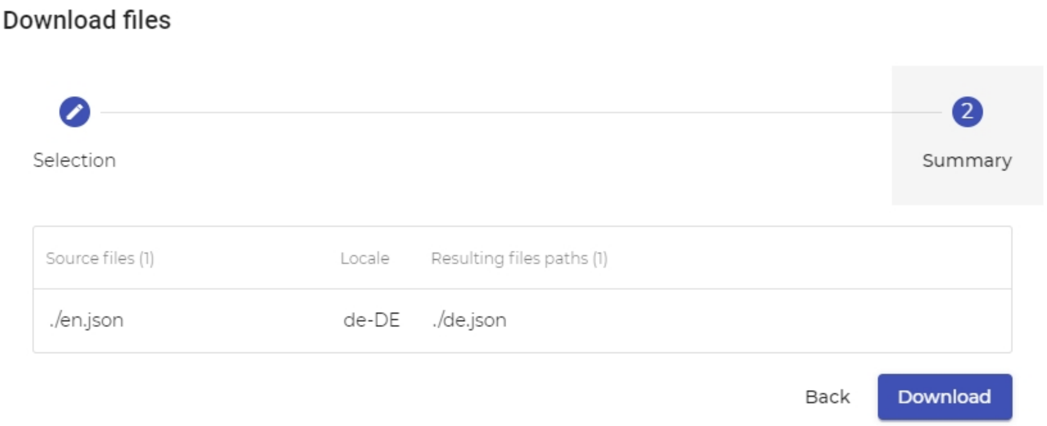
1 Selection — 2 Summary

Target files:

☑ Select all ☑ German (Germany)

□ Files with Rigi tokens
□ Source files
□ Rigi solution files

Click Next.

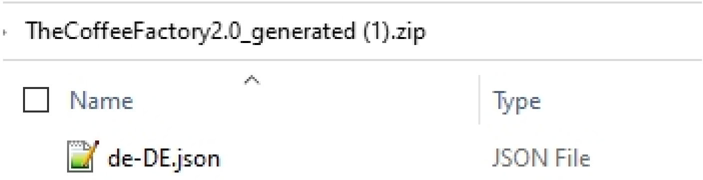


Download files

Selection — 2 Summary

Source files (1)	Locale	Resulting files paths (1)
./en.json	de-DE	./de.json

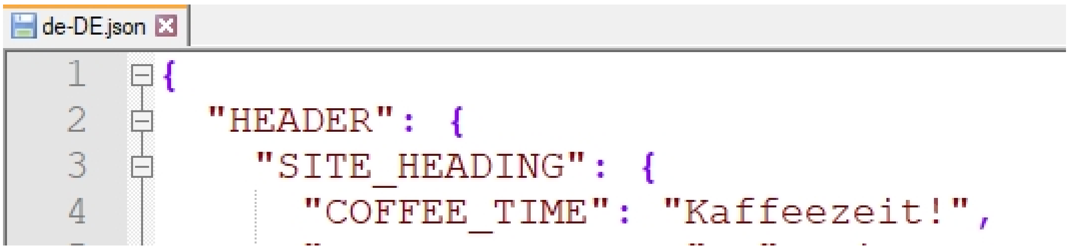
3. Click **Download**.



TheCoffeeFactory2.0_generated (1).zip

Name	Type
de-DE.json	JSON File

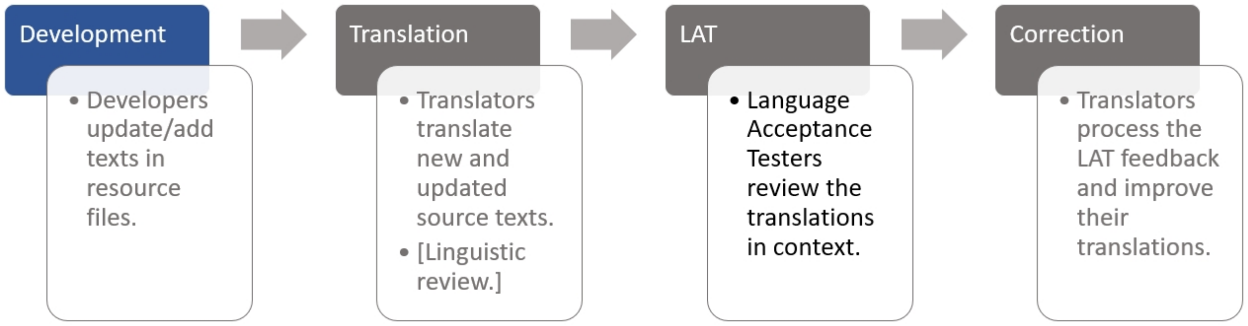
This file contains the German translations



```
{
  "HEADER": {
    "SITE_HEADING": {
      "COFFEE_TIME": "Kaffeezeit!",
```

Appendix: Process

The following figure shows the set up of the project as part of the overall localization process.


Development
• Developers update/add texts in resource files.
Translation
• Translators translate new and updated source texts.
• [Linguistic review.]
LAT
• Language Acceptance Testers review the translations in context.
Correction
• Translators process the LAT feedback and improve their translations.

Development

Developers send the complete set of resource files (e.g. json, properties) to be translated to the Project Manager.

In order to do this, the Rigi project must be set up on the Rigi server.

Translation

The Project Manager creates a project and manages that translators translate new and updated source texts. That process may include an internal linguistic review.

LAT

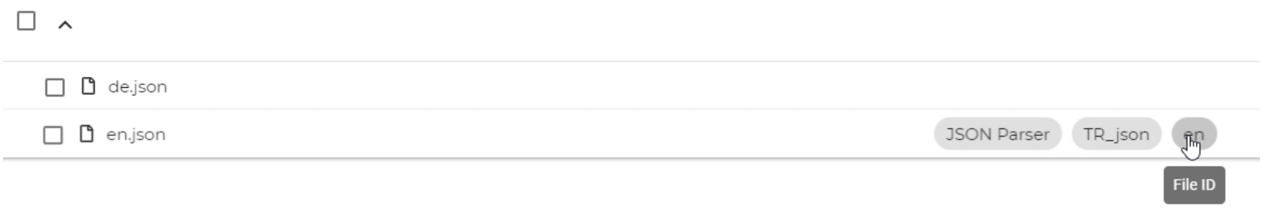
The Project Manager (you!) sets up a Language Acceptance Test. For each language a review task is created and assigned to one or more Language Testers. Each Language Tester reviews translations in context and provides feedback. That feedback may include a suggestion for a better translation.

Correction

Translators have direct access to your feedback and directly apply the corrections in their translation editor. The translators see a preview with the text highlighted.

Appendix: File ID

Each selected file in the workspace has a file identifier.



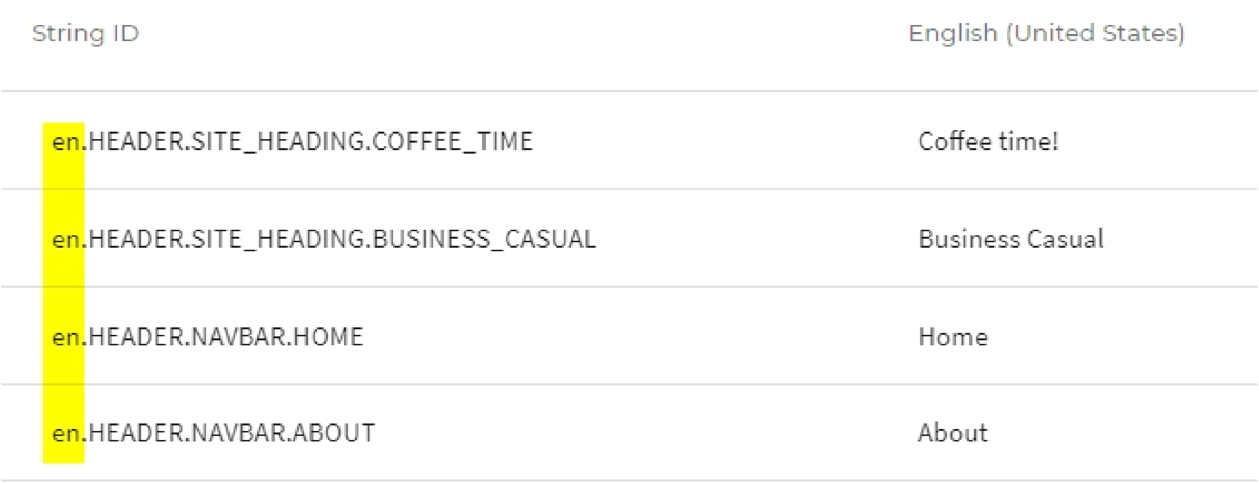
□ ^

□ de.json

□ en.json JSON Parser TR_json en

File ID

Rigi uses that ID to construct a unique Rigi identifier. For example, if the ID is en, then the string ID for each strings in that file starts with en.



String ID	English (United States)
en.HEADER.SITE_HEADING.COFFEE_TIME	Coffee time!
en.HEADER.SITE_HEADING.BUSINESS_CASUAL	Business Casual
en.HEADER.NAVBAR.HOME	Home
en.HEADER.NAVBAR.ABOUT	About

Those IDs are present as well in HTML Previews. The IDs should not be changed once assigned, because in that case strings on HTML Previews and the project may not match anymore.

Recommendations

File splitting

When developers split a file (with strings that are present in captured HTML Previews), then you should give the same File ID to both files.

File relocation and renaming

In some cases, the developers relocate files and/or rename them. When those files are uploaded to the workspace, they are not recognized. They would be recognized as new files and the existing files (and all translations!) will be removed from the project. In that case, you need to provide those files the existing file IDs.

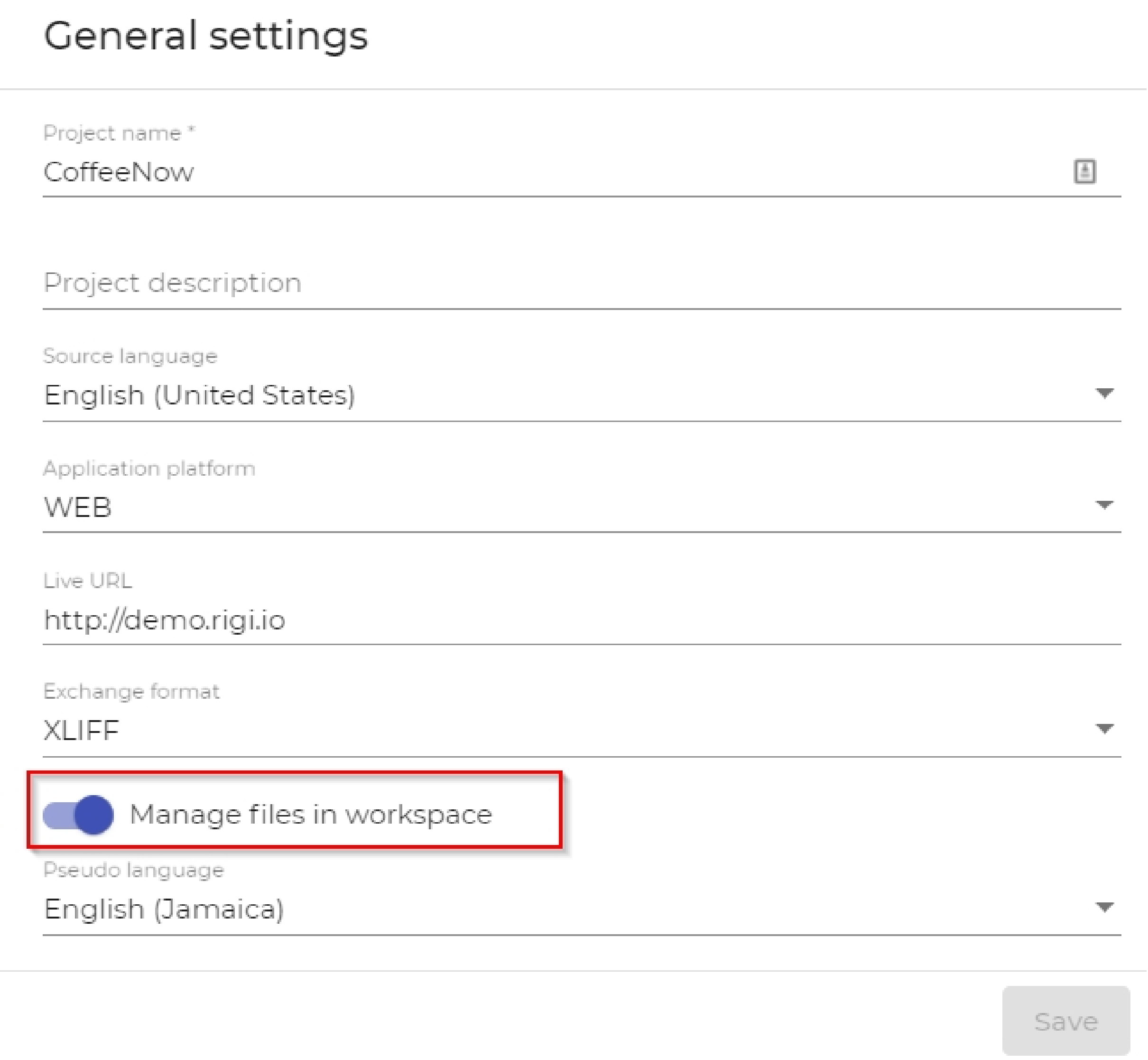
If the project contains many files, then the mass rename feature of file IDs may be useful. It enables you to export the current file paths, names and file IDs. You can rename the file IDs and import them again.

Appendix: Workflow without workspace

Default workflows use the workspace and all files are managed by the Rigi server. Some examples:

Use case	Description	Reference	Source texts required	Translations required
Translation in context in third party tool.	A third party translation tool uses HTML Previews that are stored on the Rigi server. Capturing those HTML Previews can be done on the Rigi server or using other tools. Eventually, they will be stored on the server.		Not required, but is useful to get statistics how many strings are covered by HTML Previews and which strings do not have a HTML Preview.	No
LAT	The Rigi server contains HTML Previews and source + translations. Language Testers can see HTML Previews with the translations and provide comments/suggestions	LAT How To Guides: • LAT for translators • LAT for Project Managers	Yes	Yes
Screenshots for documentation	Generate localized screenshots for documentation for a selected set of languages.		Yes	Yes

1. Open **Settings > General settings**



General settings

Project name *
CoffeeNow

Project description

Source language
English (United States)

Application platform
WEB

Live URL
http://demo.rigi.io

Exchange format
XLIFF

Manage files in workspace

Pseudo language
English (Jamaica)

2. Import the source and/or target texts directly into the string list. The following options are available:

Method	Description	Reference
Translation Uploader Tool	Developers use this tool to upload source texts and translations to the Rigi server. The tool parses the files in the local repositories, compares them with the current texts on the server, and then uploads the differences.	Translation Uploader Tool
Upload from translation tool (Rigi plugin)	Rigi plugins for some tools enable to upload texts to the Rigi server. Examples are Rigi for SDL Passolo (source + target texts) and Rigi for SDL Trados Studio	

	(target texts only).	
Rigi command line tool (rigi.exe)	The Rigi command line tool has a method to upload translations to the Rigi server.	Contact the Rigi team.
Import translations via the UI	On the Rigi server, select **Texts > Overview > Import**.	