How To translate FOLIO

- Basic Info
- Translation deployment
- Language rules
- Plural syntax
- Soft hyphen to break words

Basic Info

- Translation is done by human translators.
- If you are willing to help with string translations, join the translation team and contact Peter Murray, he will set you up on lokalise.com.
- To begin work on translations for a new language or region, contact Peter Murray.
- Currently all translators must use the web tool Lokalise (https://lokalise.com/) that shows the English text and a text field for the translation. Lokalise gives advice when the text contains placeholders and offers machine translations as suggestions.
- Do you want to use a different online or offline computer aided translation tool (CAT)? Contact Peter Murray to setup downloading and uploading XLIFF files.
- You are a developer? Read Explain internationalization and localization.

Translation deployment

- Translations are moved from Lokalise to the /translations/ directories of GitHub repositories by a script that Peter Murray runs every few days. They show up as a pull request on each stripes-* and ui-* repository, are run through the CI tests skipping the UI testing portion, and automatically merged. CI tests don't always pass and may require manual restart.
- This is a one-way process. Any manual change to locale files on GitHub will be overwritten and are lost when new locale files from Lokalise come in.
- https://folio-testing.dev.folio.org/ is the first reference environment where new or changed translations appear because it uses the master branch. Other reference environments take longer because they only use released front-end module versions, see Reference Environments for details.
- Any FOLIO release like Honeysuckle, Iris, etc. ships with the translations that come with each front-end module. Each front-end module has a different release date (Honeysuckle example). This is the time when translations are taken from module's GitHub translations directory and is different for each front-end module. A new front-end module release is needed to get more recent translation included into a FOLIO release.
- However, the sysOp of a FOLIO installation may manually update the translations of their installation by exchanging the files in the /translations/ directory from where the static files are shipped. See Backport how to do it.
- Translation keys deleted from en.json still exist on lokalise.com for older FOLIO releases. Example: A translation key is no longer in use in Iris release, but translators are still working on the Honeysuckle release where it is used.
  - TODO: Write a script that removes translation keys on lokalise.com after a grace period. Old translations will remain in the GitHub history (example). Proposal for grace period: 4 months, matching the release cycle.
  - TODO: Write a script that searches for translation keys deleted from en.json and add a comment on lokalise.com stating the last FOLIO release it is in use.
  - TODO: Write a script that checks that all translation keys in en.json are still in use in that ui module and fail a pull request if not. An exemption file, for example /translations/protected-translation-keys.json, lists translations keys that should not fail the pull request, this is for translation keys that are dynamically concatenated at run-time, that are used by other modules, or that fail at the time when the script is installed and need to be checked at a later time. The value in protected-translation-keys.json explains the reason, for example "dynamically created" or "TODO: check whether this key is still in use, remove if not".

Language rules

See Language rules in "MOTIF: The FOLIO design system" for the user experience (UX) recommendations and guidelines for the language of text in FOLIO's user interface.

Plural syntax

To support languages with multiple plural forms there is a special syntax for plurals (ICU message syntax, see https://formatjs.io/docs/core-concepts/icu-syntax/#plural-format for details).

Example 1:

<table>
<thead>
<tr>
<th>English</th>
<th>{count, number} {count, plural, one {record} other {records}} found</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>{count, number} {count, plural, one {Datensatz} other {Datensätze}} gefunden</td>
</tr>
<tr>
<td>Polish</td>
<td>znalezione {count, number} {count, plural, one {rekord} other {rekordy} few {rekordy} many {rekordów}}</td>
</tr>
</tbody>
</table>

Example 2:

<table>
<thead>
<tr>
<th>English</th>
<th>These match profile edit will affect {amount, number} associated job {amount, plural, one {profile} other {profiles}}. Please confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portuguese (Brazil)</td>
<td>Essa edição do perfil de correspondência afetará {amount, number} trabalho associado {amount, plural, one {perfil} other {perfis}}. Por favor confirme</td>
</tr>
</tbody>
</table>
Spanish (Spain) | Esta edición del perfil de coincidencia afectará a {amount, number} trabajo asociado {amount, plural, one { perfil } other { perfiles }}. Por favor confirmar

Within the curly brackets {} do not translate the key words, only translate the text in the inner curly brackets after the category key words {zero, one, two, few, many, other}, see Unicode list of languages and category key words.

**Soft hyphen to break words**

Browsers break long words that don't fit into small columns. To indicate where to split and insert the hyphen a soft hyphen can be used. The soft hyphen is invisible as long as no break at that position is needed.

This is an example that contains two soft hyphens. Use it for copy & paste: `Inventari-sierungs-status`

You see how it properly breaks when you change the browser window width.

Lokalise.com displays soft hyphens as red dots:

![filter.receivingStatus](image)

Receiving status

![Inventari-sierungs-status](image)

You cannot use this image for copy & paste, use the example a few lines above instead!

Use soft hyphens only where needed and sparingly. Reasons:

- Screen reader support is limited (for example NVDA issue).
- The lokalise.com search slot doesn't support soft hyphens: When entering the word without soft hyphens it doesn't find the word with soft hyphens. And when entering the word with soft hyphens it doesn't find the word with at least one different soft hyphen.