

DSpace 7 Translation - Internationalization (i18n) - Localization (l10n)

- [Introduction](#)
- [Getting started](#)
- [2020-05-18 Development status](#)
- [Volunteer!](#)
 - [Dutch \(nl.json5\)](#)
 - [German \(de.json5\)](#)
 - [Spanish \(es.json5\)](#)
 - [Brazilian Portuguese \(pt-BR.json5\)](#)
 - [French \(fr.json5\)](#)
 - [Finnish \(fi.json5\)](#)
 - [Latvian \(lv.json5\)](#)
 - [Polish \(pl.json5\)](#)
- [Translator documentation](#)
 - [The authoritative English master file \(en.json5\)](#)
 - [Translations \(for example, nl.json5\)](#)
 - [Starting a new language translation from scratch](#)
 - [Validating your file with the json5 validator](#)
 - [Checking your files syntax via "lint"](#)
 - [Your first translation pull request!](#)
- [Developer i18n How-to](#)
 - [Escaping](#)
 - [Syncing existing translations with changes to en.json5](#)
- [Future work](#)
 - [ICU Expressions and Pluralization](#)
- [What happened to the objective to leverage .po and gettext as new standard?](#)
 - [2019-07-25 State of development \(OLD\)](#)
 - [2019-05-20 State of development \(OLD\)](#)
 - [Grep commands for identifying keys, if we ever want to replace them with the English string again](#)
 - [Background reading](#)

Introduction

Prior to DSpace 7, the DSpace XML and JSP User interfaces had different catalogs of interface messages. Unified on a single user interface, the DSpace community is transitioning to a single catalog of interfaces messages and better tools for translators.

The DSpace community is actively seeking contributors to aid in the translation of DSpace interface messages, to ensure that DSpace 7 can benefit from the most extensive localization support in the history of the project.

Getting started

Contributing to translations only requires a (free) Github account. Head over to www.github.com and get started.

Make yourself familiar with the new json5 catalog of DSpace messages

- <https://github.com/DSpace/dspace-angular/blob/master/src/assets/i18n/en.json5>
 - This English master file is the authoritative source for the list of keys that are used in the angular interface, as well as the associated English messages
- <https://github.com/DSpace/dspace-angular/blob/master/src/assets/i18n/nl.json5>
 - As one example of the translated catalog, you will see in the dutch catalog that the English original messages are still present as comments.

The following video walks you through the contribution process:

2020-05-18 Development status

This documentation page has been updated to reflect that the language catalogs have moved from `/resources/i18n` to `/src/assets/i18n` and that `sync-i18n-files.js` has now become a typescript file.

Instructions to start the script have been updated.

Early DSpace 7 Documentation is now available at [Multilingual Support](#)

Volunteer!

Please list your name, email address alongside any of the languages to which you wish to contribute. Also feel free to join the channel #translation on the [DSpace Slack](#) for assistance and discussion around DSpace 7 translations.

Dutch (nl.json5)

[Marina Muilwijk](#)

German (de.json5)

[Claudia Jürgen](#)

Spanish (es.json5)

[Maria Fernanda Ruiz](#)

Eva Braña Arvo Consultores for Version 7.3)

Emilio Lorenzo (Arvo Consultores for Version 7.3)

Brazilian Portuguese (pt-BR.json5)

[Vitor Silvério Rodrigues](#)

French (fr.json5)

[David Cavrenne \[Atmire\]](#)

[Marie-Hélène Vézina](#)

Finnish (fi.json5)

[Reeta Kuuskoski](#)

Latvian (lv.json5)

[Denijs Balodis](#)

Polish (pl.json5)

[Micha Dykas](#) (PCG Academia)

Translator documentation

The authoritative English master file (en.json5)

The catalogs can be found in [src/assets/i18n](#)

en.json5 example section

```
"submission.workflow.tasks.claimed.reject.reason.submit": "Reject item",

"submission.workflow.tasks.claimed.reject.reason.title": "Reason",

"submission.workflow.tasks.claimed.reject.submit": "Reject",

"submission.workflow.tasks.claimed.reject_help": "If you have reviewed the item and found it is <strong>not</strong> suitable for inclusion in the collection, select \"Reject\". You will then be asked to enter a message indicating why the item is unsuitable, and whether the submitter should change something and resubmit.",
```

Translations (for example, nl.json5)

In order to allow automated syncing with changes in the English master catalog, and to make it clear to all translators what the original source of the message was, the English original is copied as a comment into the translated files

nl.json5 example section

```
// "404.link.home-page": "Take me to the home page",
"404.link.home-page": "Terug naar de homepagina",

// "404.page-not-found": "page not found",
"404.page-not-found": "Pagina niet gevonden",

// "admin.registries.bitstream-formats.create.failure.content": "An error occurred while creating the new
bitstream format.",
// TODO New key - Add a translation
"admin.registries.bitstream-formats.create.failure.content": "An error occurred while creating the new
bitstream format.",
```

When new keys are introduced, TODO messages are automatically added to the catalogs of other languages.

Starting a new language translation from scratch

1. Create an empty file in resources/i18n and give it the [two letter iso code of your language](#) as the filename, for example nl.json5 or fr.json5
2. Preload your new catalog with all the messages in the en.json5 catalog by executing the sync-i18n-files.ts script, for which you can find more detailed documentation lower on this page. Assuming you are in /src/assets/i18n, you can execute:

```
yarn run sync-i18n -s en.json5 -i -t fr.json5
```

or

```
ts-node --project ../../../../tsconfig.ts-node.json ../../../../scripts/sync-i18n-files.ts -s en.json5 -i -t fr.json5
```

where en.json5 is the "source" language file where the original keys will be retrieved, and fr.json5 is the catalog of the new language for which you want to start a translation.

The file should now be filled with all of the English messages, that are now present both as comments, as well as message, for you to translate.

3. Translate as many messages as you can. For every message you translate, remove the TODO comment from the catalog.
4. Activate the new language in [./src/config/default-app-config.ts](#) (look for the "LangConfig[]" array)

```
languages: LangConfig[] = [
  ...
  { code: 'nl', label: 'Nederlands', active: true },
  ...
];
```

Where code is the two letter code for your language, also used as the filename for your translated catalog.

Label is the name of your language, written in that language.

Active set to true, so the language is visible in the language selector on the frontpage.

Validating your file with the json5 validator

If you have the DSpace 7 UI running locally, including the yarn tooling to start it up, there is a handy command line tool to validate your new translation. For example, to validate the Spanish file "es.json5", you can execute:

json5 validate

```
yarn json5 --validate ./src/assets/i18n/es.json5
```

Examples of possible errors you might get:

```
JSON5: invalid character 'R' at 1574:179
```

1574 indicates the line number, and 179 is the 179th character on that line.

This particular mistake occurred where there was an extra space between \ and " in the \" combination used to escape the "

Checking your files syntax via "lint"

Lint is a tool we use to ensure files are correctly formatted, and have no extra characters (e.g. extra spaces at the end of each line can make the file large in size, which take longer for users to download). This simple command will check your file's syntax (run this from the root folder):

```
yarn lint --lint-file-patterns src/assets/i18n/*.json5
```

(NOTE: Warnings shown by lint can be ignored. Errors are the only thing that require cleanup)

If that throws errors, most can be automatically cleaned up by adding "--fix" to the same command:

```
yarn lint --lint-file-patterns src/assets/i18n/*.json5 --fix
```

NOTE: currently, "--fix" is unable to add missing commas on the end of each line. Every new line (which is not a comment) MUST end in a comma. However, if you have a missing comma, you'll see an error like this:

```
# This error is a sign that you have a missing comma on the line just BEFORE [some translation key]
Parsing error: Unexpected token [some translation key]
```

Your first translation pull request!

Once you have your first translation, in your dspace-angular repository on your account on Github, you can send these translations in as a pull request. Following screenshots provide clarification:

Overview of your branches, where you can start the pull request. This is at <https://github.com/bram-atmire/dspace-angular/branches> where "bram-atmire" is changed by your own user name.

🔗 [bram-atmire / dspace-angular](#) Unwatch 1 ★ Star 0 🔗 Fork 65

forked from [DSpace/dspace-angular](#)

Code Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Overview Yours Active Stale All branches

Default branch

master Updated 15 months ago by artlowel Default Change default branch

Your branches

Branch Name	Updated	By	Commits	Actions
Issue570-Derivative-Portuguese-language	Updated 11 days ago	by bram-atmire	0 2002	New pull request 🗑️
PR557-Partial-Dutch-Translation	Updated 11 days ago	by bram-atmire	0 1999	New pull request 🗑️
placeholder-catalogs	Updated 4 months ago	by bram-atmire	0 1483	New pull request 🗑️
PR515-Spanish	Updated 4 months ago	by bram-atmire	0 1467	New pull request 🗑️
issue-331-migrate-json-language-catalogs-to-po	Updated 9 months ago	by bram-atmire	0 771	New pull request 🗑️

[View more of your branches >](#)

Active branches

Branch Name	Updated	By	Commits	Actions
Issue570-Derivative-Portuguese-language	Updated 11 days ago	by bram-atmire	0 2002	New pull request 🗑️
PR557-Partial-Dutch-Translation	Updated 11 days ago	by bram-atmire	0 1999	New pull request 🗑️

When you make a pull request, make sure it goes to the "base repository" DSpace/dspace-angular, into the master branch

[DSpace / dspace-angular](#) Unwatch 45 ★ Star 30 🔗 Fork 65

Code Issues 92 Pull requests 21 Actions Projects 0 Wiki Security Insights Settings

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

🔗 base repository: [DSpace/dspace-angular](#) base: [master](#) ↔

head repository: [bram-atmire/dspace-angular](#) compare: [Issue570-Derivative-Portug...](#)

✓ **Able to merge.** These branches can be automatically merged.

If you experience any difficulties with this, people are ready to help you on #translation in the [DuraSpace slack](#)

Developer i18n How-to

When you create new keys, update existing keys or the meaning of existing messages, keep the following in mind.

Escaping

In the message catalogs, double quotes have to be escaped with \

```
"submission.workflow.tasks.claimed.approve_help": "If you have reviewed the item and it is suitable for inclusion in the collection, select \"Approve\"."
```

Syncing existing translations with changes to en.json5

Whenever you make one of following changes to en.json5:

- Introduce new keys & messages
- Change an existing key
- Change an existing message

Please run the node script: /scripts/sync-i18n-files.ts.

There are two ways to run this script:

- yarn run sync-i18n
- ts-node --project ../../tsconfig.ts-node.json ../../scripts/sync-i18n-files.ts ==> assuming that you execute this while you are in /src/assets/i18n

The results of your changes will be reflected in the catalogs of the other languages, so translators can pick up the work to:

- Provide translations for the new keys you just introduced
- Update existing translations in case you changed the message for an existing key

By default, the execution of sync-i18n-files.ts will:

- look at en.json5 as the authoritative **source** file
- will look at all catalogs present in /src/assets/i18n/ as **target** files to sync up with the latest changes in source.
- will execute these changes **in-place**, meaning that no backup etc is taken of the translation before the sync

If you want to alter this behaviour, you can:

- specify a different source file using **-s**
- specify a different target file using **-t**
- specify another output file using **-o** in case you want to avoid in place edit

Future work

ICU Expressions and Pluralization

Instead of relying on different messages for singular and plural forms, the community hopes to conform the messages to the [ICU Message Format](#) as specified in the [CLDR Plural rules](#)

What happened to the objective to leverage .po and gettext as new standard?

Originating in the Linux world, [the GNU gettext tools and the PO file format](#) is also the backbone of localization support in [Wordpress](#), [django](#) and [Drupal](#)

The Angular i18n framework we use, NGX translate, has a 3rd party PO file loader: <https://github.com/biesbjerg/ngx-translate-po-http-loader>

Even though the community was initially very optimistic about its potential and the transition to .po and gettext, the major deal breaker was the absence of support for gettext message context (msgctxt), that would allow a translator to translate a key like "Home" into different words in the target language, depending on the context.

The initial ambitions to use the English string as the key itself, and abandon intermediate keys, was also problematic, as we hit a big number of areas in the code where keys were built up programmatically.

As a result, the community settled for:

- JSON5 as the format for the message catalog
- Reverting to a flat list of keys, instead of a hierarchical tree. This now makes it possible again to search on a particular key, which was not possible anymore in the hierarchical format.

2019-07-25 State of development (OLD)

The key challenges that are still being tackled are:

- Compatibility between the PO loader for NGX and Angular 7 cfr <https://github.com/biesbjerg/ngx-translate-po-http-loader/issues/29>
- Support for pluralization
- Support for the translation of a specific English message, into different messages in another language, depending on the context.

2019-05-20 State of development (OLD)

As part of preview release 1, the developers are still using en.json catalogs. Once [Pull Request 366](#) is accepted, the migration to the new .POT and .PO standard files will be official.

As long as DSpace 7 is still in development, it is expected that the dspace.pot catalog, as well as the different translations, will continue to be extended and evolved.

Together, we aim to release as many, as complete translations as possible, as part of the official DSpace 7.0 release.

Grep commands for identifying keys, if we ever want to replace them with the English string again

Because we are currently **keeping message keys**, there is no immediate use for identification of keys to replace with the English strings. But for future use, this might still come in handy:

If you execute following command in the angular source directory, you get a list of keys that have not yet been replaced.

Grep command for identifying keys that have not yet been replaced

```
grep -snRHiiE "'.*\.[^\s]+\.[^']+ ' \|| translate" *
```

Sample output looks like:

```
app/+community-page/delete-community-page/delete-community-page.component.html:5: <h2 id="header" class="border-bottom pb-2">{{ 'community.delete.head' | translate
app/+community-page/delete-community-page/delete-community-page.component.html:7: <p class="pb-2">{{ 'community.delete.text' | translate:{ dso: dso.name } }}</p>
app/+community-page/delete-community-page/delete-community-page.component.html:12: <button class="btn btn-primary" (click)="onCancel(dso)">{{ 'community.delete.cancel' | translate}}
app/+community-page/edit-community-page/edit-community-page.component.html:4: <h2 id="header" class="border-bottom pb-2">{{ 'community.edit.head' | translate }}</h2>
```

Background reading

[The PO Format](#)

[GNU gettext utilities](#)