

Version: 3.0.0

i18n - Using git

A **possible translation strategy** is to **version control the translation files** with Git (or any other [VCS](#)).

Tradeoffs

This strategy has advantages:

- **Easy to get started:** just commit the `i18n` folder to Git
- **Easy for developers:** Git, GitHub and pull requests are mainstream developer tools
- **Free** (or without any additional cost, assuming you already use Git)
- **Low friction:** does not require signing up to an external tool
- **Rewarding:** contributors are happy to have a nice contribution history

Using Git also present some shortcomings:

- **Hard for non-developers:** they do not master Git and pull-requests
- **Hard for professional translators:** they are used to SaaS translation software and advanced features
- **Hard to maintain:** you have to keep the translated files **in sync** with the untranslated files

NOTE

Some **large-scale technical projects** (React, Vue.js, MDN, TypeScript, Nuxt.js, etc.) use Git for translations.

Refer to the [Docusaurus i18n RFC](#) for our notes and links studying these systems.

Initialization

This is a walk-through of using Git to translate a newly initialized English Docusaurus website into French, and assume you already followed the [i18n tutorial](#).

Prepare the Docusaurus site

Initialize a new Docusaurus site:

```
npx create-docusaurus@latest website classic
```

Add the site configuration for the French language:

docusaurus.config.js

```
export default {
  i18n: {
    defaultLocale: 'en',
    locales: ['en', 'fr'],
  },
  themeConfig: {
    navbar: {
      items: [
        // ...
        {
          type: 'localeDropdown',
          position: 'left',
        },
        // ...
      ],
    },
  },
  // ...
};
```

Translate the homepage:

src/pages/index.js

```
import React from 'react';
import Translate from '@docusaurus/Translate';
import Layout from '@theme/Layout';

export default function Home() {
  return (
    <Layout>
      <h1 style={{margin: 20}}>
        <Translate description="The homepage main heading">
          Welcome to my Docusaurus translated site!
        </Translate>
      </h1>
    </Layout>
  );
}
```

```
    </Translate>
  </h1>
</Layout>
);
}
```

Initialize the `i18n` folder

Use the `write-translations` CLI command to initialize the JSON translation files for the French locale:

Yarn

```
yarn write-translations --locale fr
```

```
1 translations written at i18n/fr/code.json
11 translations written at i18n/fr/docusaurus-theme-classic/footer.json
4 translations written at i18n/fr/docusaurus-theme-classic/navbar.json
3 translations written at i18n/fr/docusaurus-plugin-content-
docs/current.json
```

TIP

Use the `--messagePrefix '(fr) '` option to make the untranslated strings stand out.

`Hello` will appear as `(fr) Hello` and makes it clear a translation is missing.

Copy your untranslated Markdown files to the French folder:

```
mkdir -p i18n/fr/docusaurus-plugin-content-docs/current
cp -r docs/** i18n/fr/docusaurus-plugin-content-docs/current

mkdir -p i18n/fr/docusaurus-plugin-content-blog
cp -r blog/** i18n/fr/docusaurus-plugin-content-blog

mkdir -p i18n/fr/docusaurus-plugin-content-pages
cp -r src/pages/**/*.md i18n/fr/docusaurus-plugin-content-pages
cp -r src/pages/**/*.mdx i18n/fr/docusaurus-plugin-content-pages
```

Add all these files to Git.

Translate the files

Translate the Markdown and JSON files in `i18n/fr` and commit the translation.

You should now be able to start your site in French and see the translations:

Yarn

```
yarn run start --locale fr
```

You can also build the site locally or on your CI:

Yarn

```
yarn build  
# or  
yarn build --locale fr
```

Repeat

Follow the same process for each locale you need to support.

Maintenance

Keeping translated files **consistent** with the originals **can be challenging**, in particular for Markdown documents.

Markdown translations

When an untranslated Markdown document is edited, it is **your responsibility to maintain the respective translated files**, and we unfortunately don't have a good way to help you do so.

To keep your translated sites consistent, when the `website/docs/doc1.md` doc is edited, you need **backport these edits** to `i18n/fr/docusaurus-plugin-content-docs/current/doc1.md`.

JSON translations

To help you maintain the JSON translation files, it is possible to run again the [write-translations](#) CLI command:

Yarn

```
yarn write-translations --locale fr
```

New translations will be appended, and existing ones will not be overridden.



TIP

Reset your translations with the `--override` option.

Localize edit URLs

When the user is browsing a page at `/fr/doc1`, the edit button will link by default to the unlocalized doc at `website/docs/doc1.md`.

Your translations are on Git, and you can use the `editLocalizedFiles: true` option of the docs and blog plugins.

The edit button will link to the localized doc at `i18n/fr/docusaurus-plugin-content-docs/current/doc1.md`.

 [Edit this page](#)

Last updated on Oct 31, 2023 by Sébastien Lorber