Version: 2.4.0

On this page

i18n - Using Crowdin

The i18n system of Docusaurus is **decoupled from any translation software**.

You can integrate Docusaurus with the **tools and SaaS of your choice**, as long as you put the **translation files at the correct location**.

We document the usage of Crowdin, as **one** possible **integration example**.

⚠ CAUTION

This is **not an endorsement of Crowdin** as the unique choice to translate a Docusaurus site, but it is successfully used by Facebook to translate documentation projects such as Jest, Docusaurus, and ReasonML.

Refer to the **Crowdin documentation** and **Crowdin support** for help.

TIP

Use this **community-driven GitHub discussion** to discuss anything related to Docusaurus + Crowdin.

Crowdin overview

Crowdin is a translation SaaS, offering a free plan for open-source projects.

We recommend the following translation workflow:

- **Upload sources** to Crowdin (untranslated files)
- Use Crowdin to **translate the content**
- **Download translations** from Crowdin (localized translation files)

Crowdin provides a CLI to **upload sources** and **download translations**, allowing you to automate the translation process.

The `crowdin.yml` configuration file is convenient for Docusaurus, and permits to **download the localized translation files at the expected location** (in `i18n/[locale]/..`).

Read the **official documentation** to know more about advanced features and different translation workflows.

Crowdin tutorial

This is a walk-through of using Crowdin to translate a newly initialized English Docusaurus website into French, and assume you already followed the i18n tutorial.

The end result can be seen at docusaurus-crowdin-example.netlify.app (repository).

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

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

Translate the homepage:

src/pages/index.js

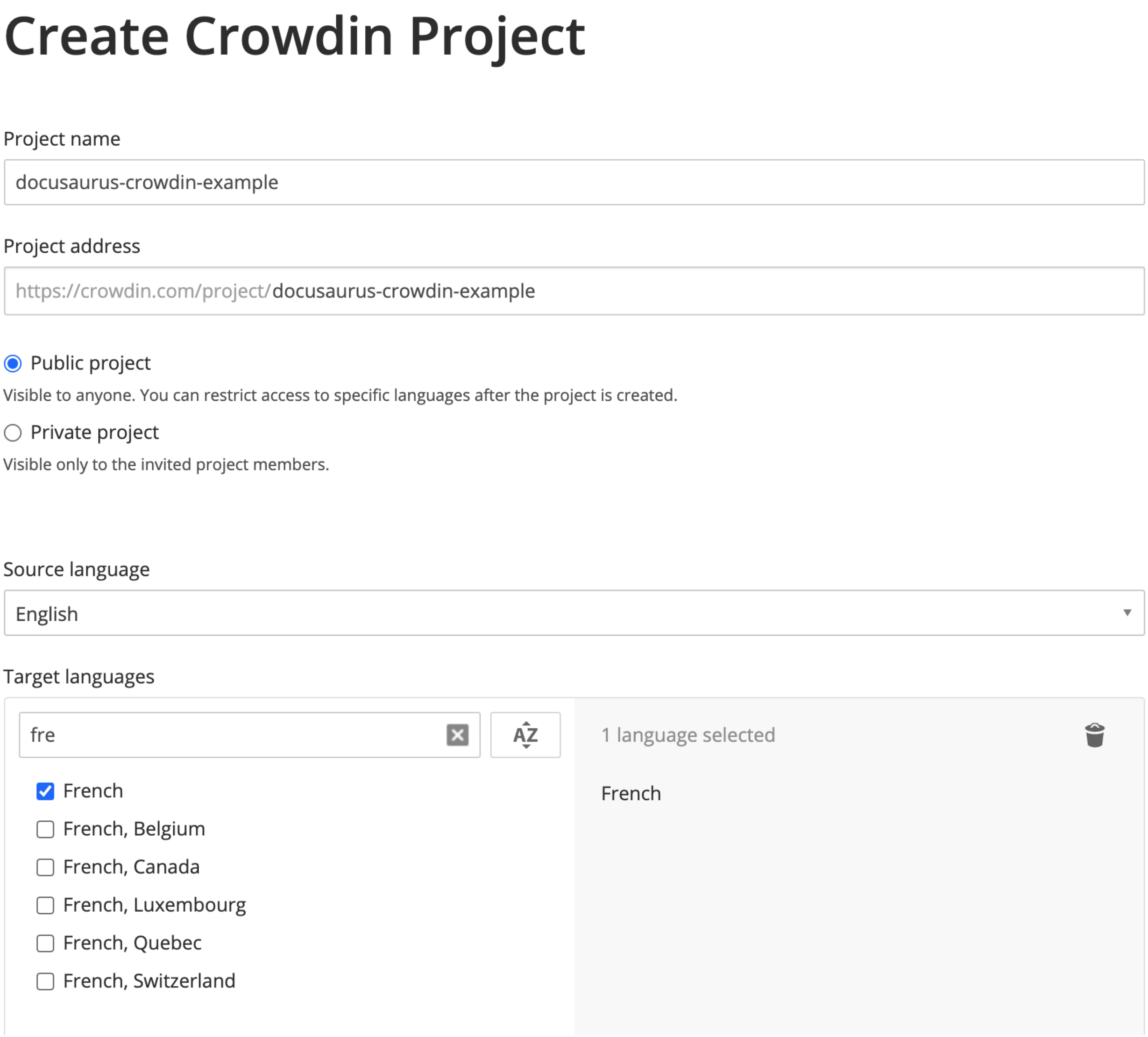
```
import React from 'react';
import Translate from '@docusaurus/Translate';
import Layout from '@theme/Layout';

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>
  );
}
```

Create a Crowdin project

Sign up on Crowdin, and create a project.

Use English as the source language, and French as the target language.



Create Crowdin Project

Project name

docusaurus-crowdin-example

Project address

https://crowdin.com/project/docusaurus-crowdin-example

● Public project

Visible to anyone. You can restrict access to specific languages after the project is created.

○ Private project

Visible only to the invited project members.

Source language

English ▾

Target languages

fre ☒ AZ

☑ French
☐ French, Belgium
☐ French, Canada
☐ French, Luxembourg
☐ French, Quebec
☐ French, Switzerland

1 language selected

French

Your project is created, but it is empty for now. We will upload the files to translate in the next steps.

Create the Crowdin configuration

This configuration (doc) provides a mapping for the Crowdin CLI to understand:

- Where to find the source files to upload (JSON and Markdown)
- Where to download the files after translation (in `i18n/[locale]`)

Create `crowdin.yml` in `website`:

crowdin.yml

```yaml
project_id: '123456'
api_token_env: CROWDIN_PERSONAL_TOKEN
preserve_hierarchy: true
files:
  # JSON translation files
  - source: /i18n/en/**/*
    translation: /i18n/%two_letters_code%/**/%original_file_name%
  # Docs Markdown files
  - source: /docs/**/*
    translation: /i18n/%two_letters_code%/docusaurus-plugin-content-docs/current/**/%original_file_name%
  # Blog Markdown files
  - source: /blog/**/*
    translation: /i18n/%two_letters_code%/docusaurus-plugin-content-blog/**/%original_file_name%
```

Crowdin has its own syntax for declaring source/translation paths:

- `**/*`: everything in a subfolder
- `%two_letters_code%`: the 2-letters variant of Crowdin target languages (`fr` in our case)
- `**/%original_file_name%`: the translations will preserve the original folder/file hierarchy

INFO

The Crowdin CLI warnings are not always easy to understand.

We advise to:

- change one thing at a time
- re-upload sources after any configuration change
- use paths starting with `/` (`./` does not work)
- avoid fancy globbing patterns like `/docs/**/*.(md|mdx)` (does not work)

Access token

The `api_token_env` attribute defines the **env variable name** read by the Crowdin CLI.

You can obtain a `Personal Access Token` on your personal profile page.

TIP

You can keep the default value `CROWDIN_PERSONAL_TOKEN`, and set this environment variable and on your computer and on the CI server to the generated access token.

CAUTION

A Personal Access Tokens grant **read-write access to all your Crowdin projects**.

You should **not commit** it, and it may be a good idea to create a dedicated **Crowdin profile for your company** instead of using a personal account.

Other configuration fields

- `project_id`: can be hardcoded, and is found on `https://crowdin.com/project/<MY_PROJECT_NAME>/settings#api`
- `preserve_hierarchy`: preserve the folder's hierarchy of your docs on Crowdin UI instead of flattening everything

Install the Crowdin CLI

This tutorial uses the CLI version `3.5.2`, but we expect `3.x` releases to keep working.

Install the Crowdin CLI as an npm package to your Docusaurus site:

npm
Yarn
pnpm

```
npm install @crowdin/cli@3
```

Add a `crowdin` script:

package.json

```
{
  "scripts": {
    // ...
    "write-translations": "docusaurus write-translations",
    "crowdin": "crowdin"
  }
}
```

Test that you can run the Crowdin CLI:

npm
Yarn
pnpm

```
npm run crowdin -- --version
```

Set the `CROWDIN_PERSONAL_TOKEN` env variable on your computer, to allow the CLI to authenticate with the Crowdin API.

TIP

Temporarily, you can hardcode your personal token in `crowdin.yml` with `api_token: 'MY-TOKEN'`.

Upload the sources

Generate the JSON translation files for the default language in `website/i18n/en`:

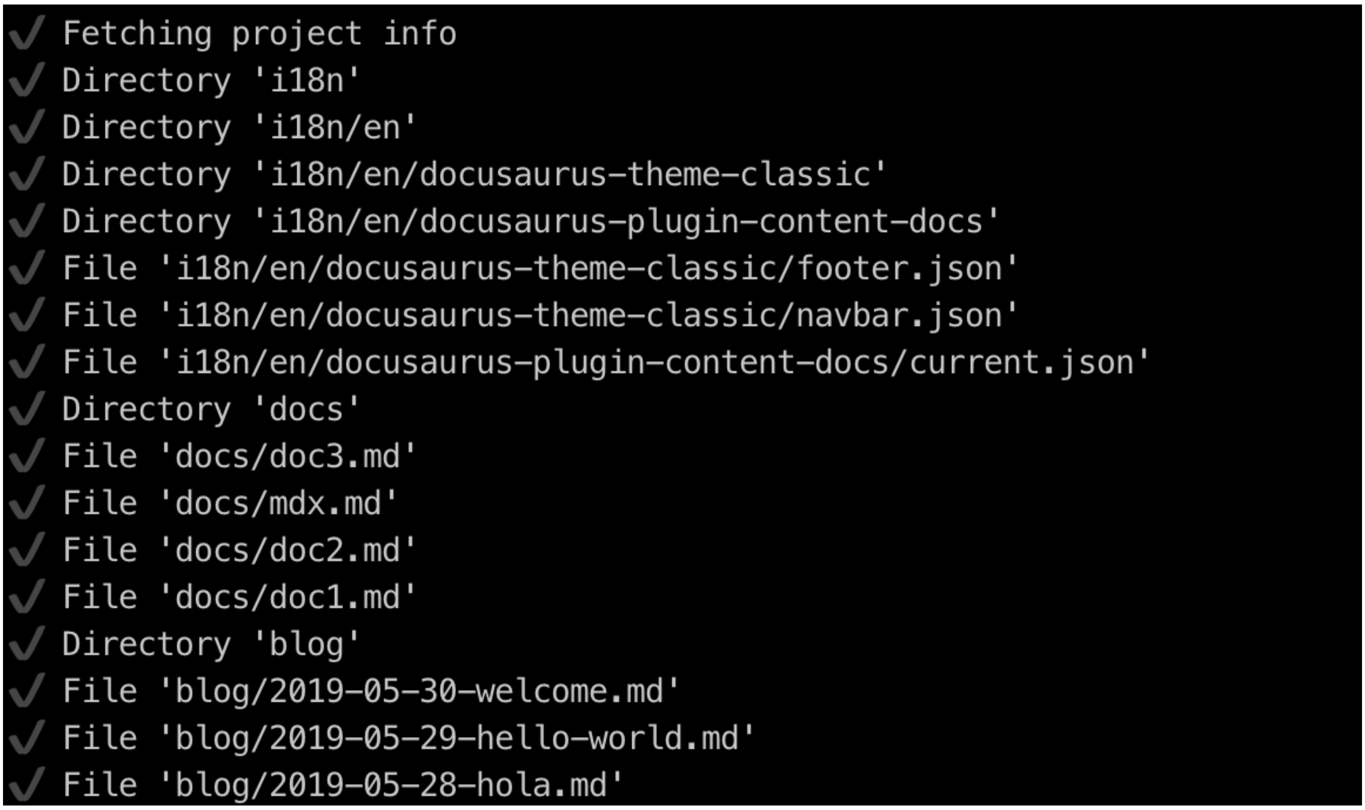
npm
Yarn
pnpm

```
npm run write-translations
```

Upload all the JSON and Markdown translation files:

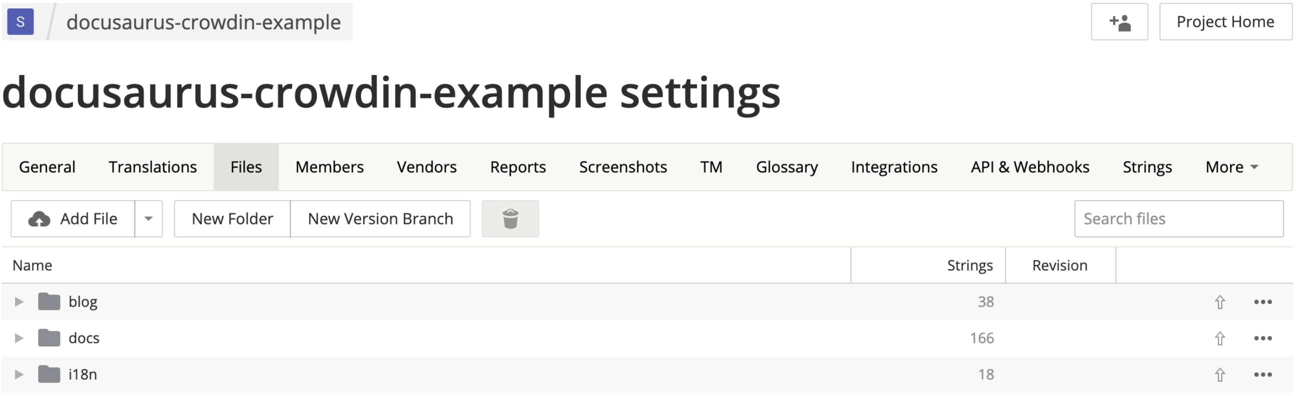
npm
Yarn
pnpm

```
npm run crowdin upload
```



```
✓ Fetching project info
✓ Directory 'i18n'
✓ Directory 'i18n/en'
✓ Directory 'i18n/en/docusaurus-theme-classic'
✓ Directory 'i18n/en/docusaurus-plugin-content-docs'
✓ File 'i18n/en/docusaurus-theme-classic/footer.json'
✓ File 'i18n/en/docusaurus-theme-classic/navbar.json'
✓ File 'i18n/en/docusaurus-plugin-content-docs/current.json'
✓ Directory 'docs'
✓ File 'docs/doc3.md'
✓ File 'docs/mdx.md'
✓ File 'docs/doc2.md'
✓ File 'docs/doc1.md'
✓ Directory 'blog'
✓ File 'blog/2019-05-30-welcome.md'
✓ File 'blog/2019-05-29-hello-world.md'
✓ File 'blog/2019-05-28-hola.md'
```

Your source files are now visible on the Crowdin interface:
`https://crowdin.com/project/<MY_PROJECT_NAME>/settings#files`



S / docusaurus-crowdin-example

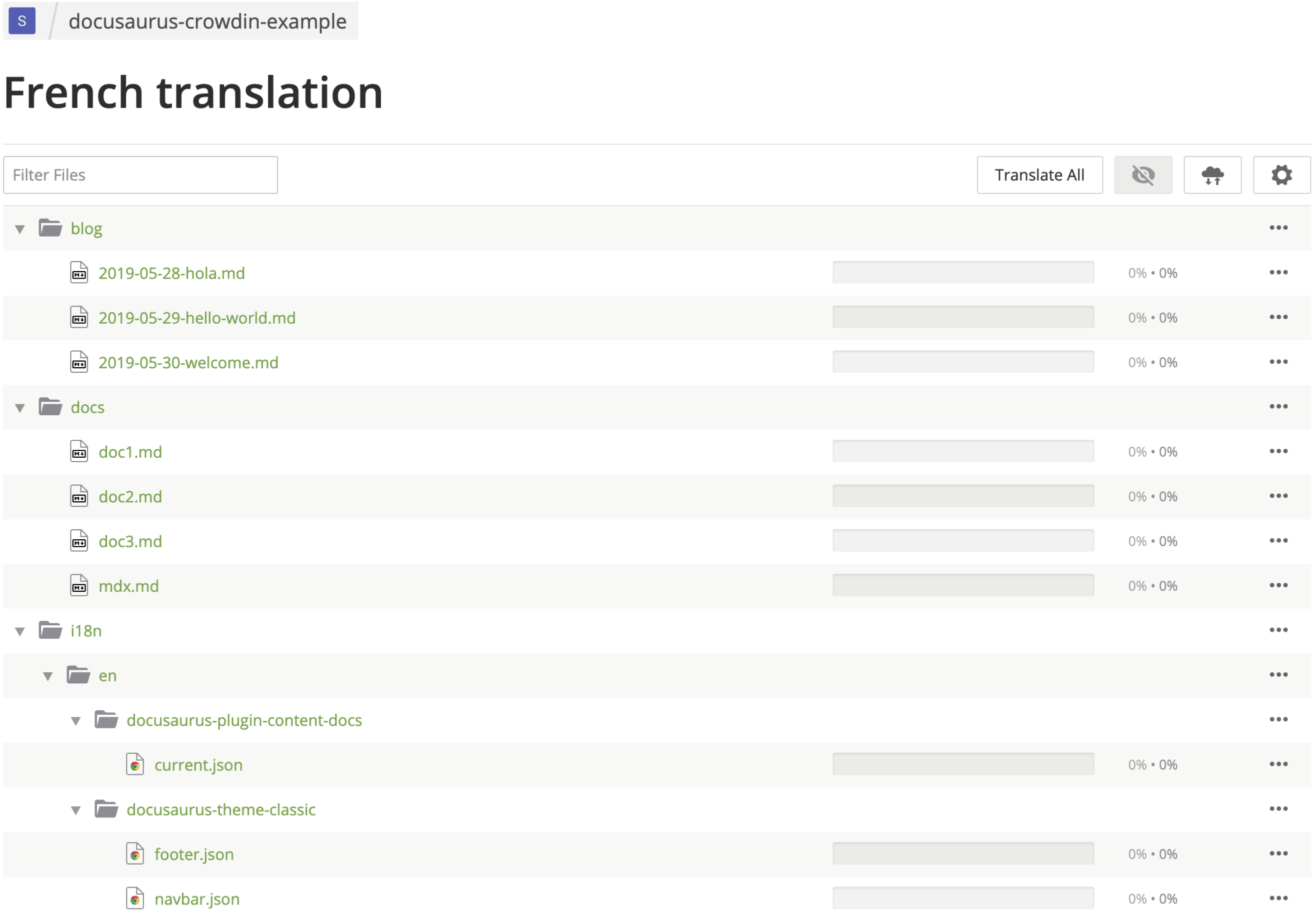
Project Home

docusaurus-crowdin-example settings

Name	Strings	Revision
blog	38	
docs	166	
i18n	18	

Translate the sources

On `https://crowdin.com/project/<MY_PROJECT_NAME>`, click on the French target language.

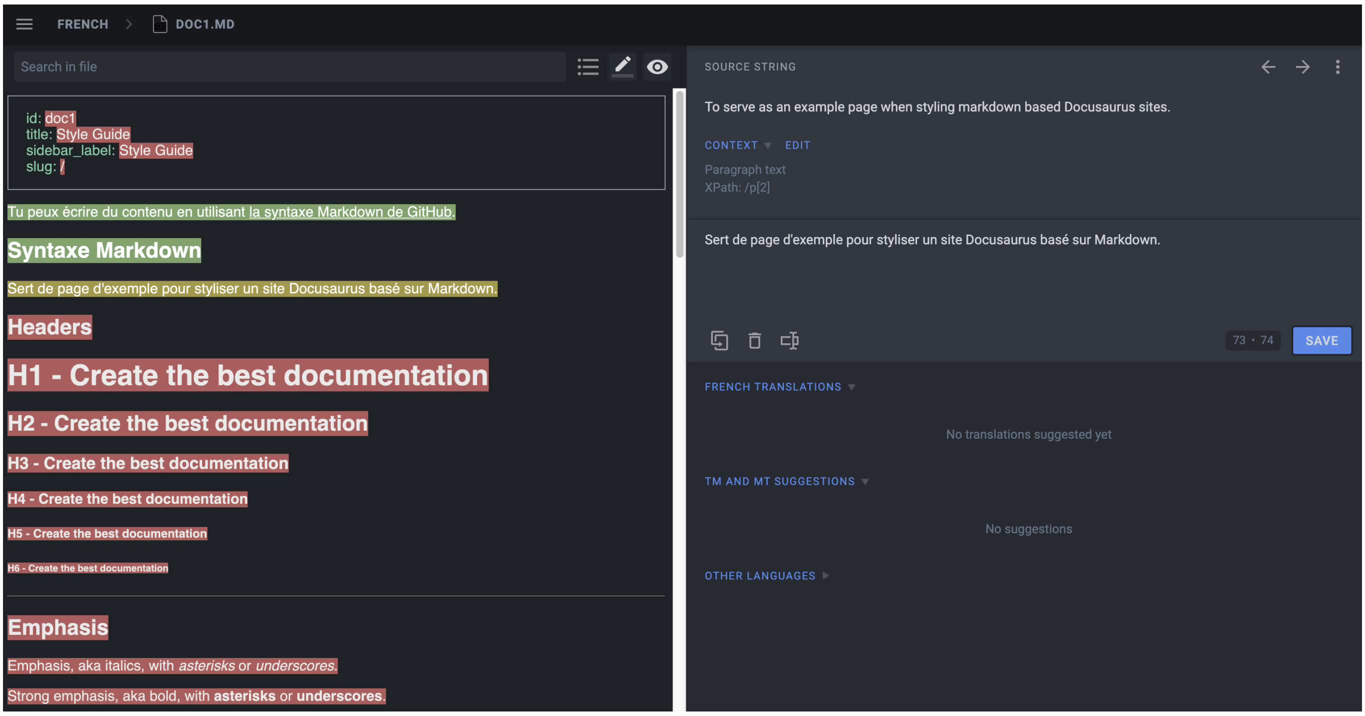


S / docusaurus-crowdin-example

French translation

Name	Progress
blog	
2019-05-28-hola.md	0% • 0%
2019-05-29-hello-world.md	0% • 0%
2019-05-30-welcome.md	0% • 0%
docs	
doc1.md	0% • 0%
doc2.md	0% • 0%
doc3.md	0% • 0%
mdx.md	0% • 0%
i18n	
en	
docusaurus-plugin-content-docs	
current.json	0% • 0%
docusaurus-theme-classic	
footer.json	0% • 0%
navbar.json	0% • 0%

Translate some Markdown files.



id: doc1
title: Style Guide
sidebar_label: Style Guide
slug: /

Tu peux écrire du contenu en utilisant la syntaxe Markdown de GitHub.

Syntaxe Markdown

Sert de page d'exemple pour styliser un site Docusaurus basé sur Markdown.

Headers

H1 - Create the best documentation

H2 - Create the best documentation

H3 - Create the best documentation

H4 - Create the best documentation

H5 - Create the best documentation

H6 - Create the best documentation

Emphasis

Emphasis, aka italics, with *asterisks* or *underscores*.

Strong emphasis, aka bold, with **asterisks** or **underscores**.

To serve as an example page when styling markdown based Docusaurus sites.

Paragraph text
XPath: /p[2]

Sert de page d'exemple pour styliser un site Docusaurus basé sur Markdown.

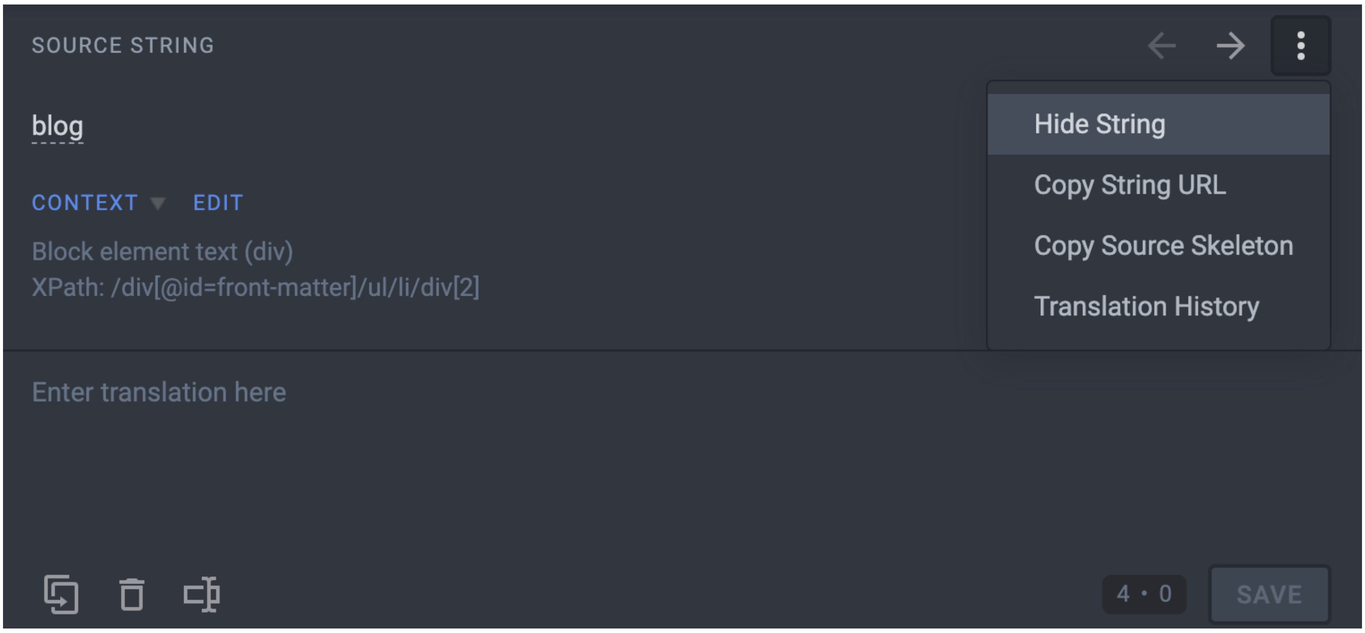
No translations suggested yet

No suggestions

TIP

Use `Hide String` to make sure translators **don't translate things that should not be**:

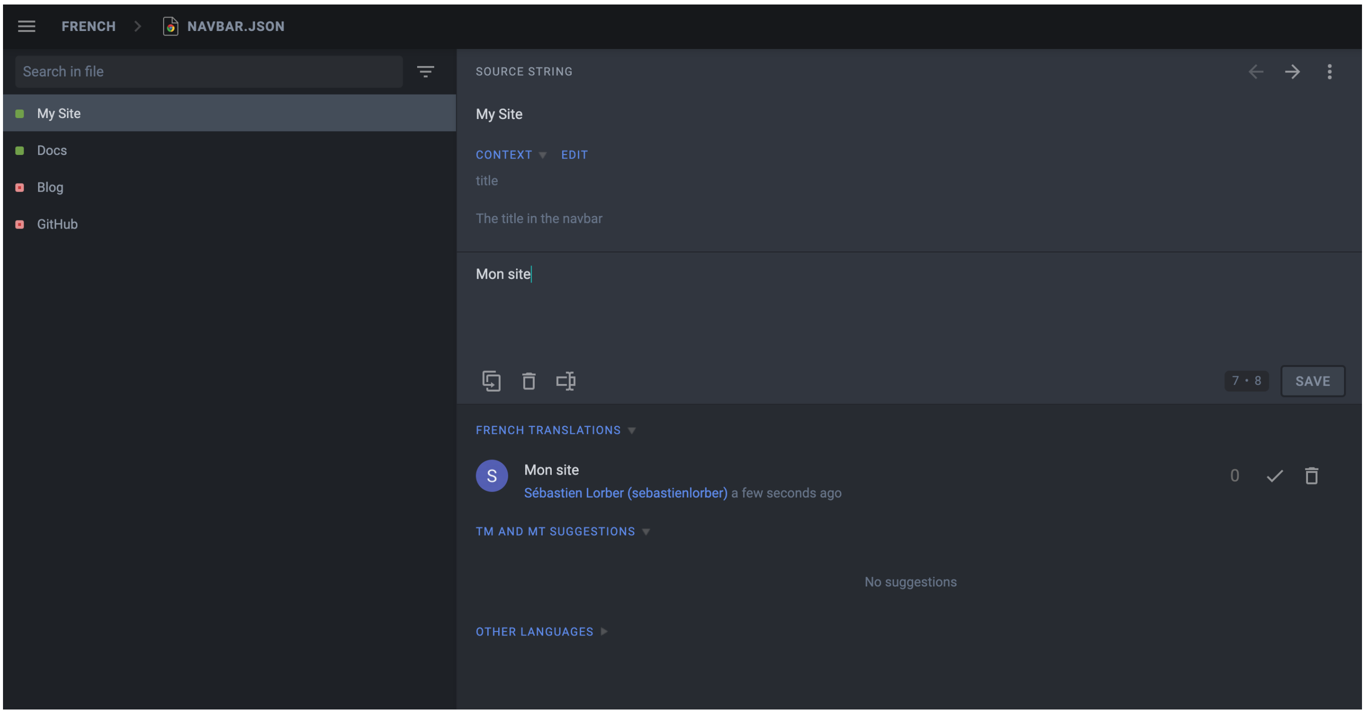
- Front matter: `id`, `slug`, `tags` ...
- Admonitions: `:::`, `:::note`, `:::tip` ...



blog

Block element text (div)
XPath: /div[@id=front-matter]/ul/li/div[2]

Translate some JSON files.



🟩 My Site
🟩 Docs
🟥 Blog
🟥 GitHub

My Site

title

The title in the navbar

Mon site

7 · 8

S Mon site
Sébastien Lorber (sebastienlorber) a few seconds ago
0

No suggestions

INFO

The `description` attribute of JSON translation files is visible on Crowdin to help translate the strings.

TIP

[Pre-translate](#) your site, and **fix pre-translation mistakes manually** (enable the Global Translation Memory in settings first).

Use the `Hide String` feature first, as Crowdin is pre-translating things too optimistically.

Download the translations

Use the Crowdin CLI to download the translated JSON and Markdown files.

npm
Yarn
pnpm

```
npm run crowdin download
```

The translated content should be downloaded in `i18n/fr`.

Start your site on the French locale:

npm
Yarn
pnpm

```
npm run start -- --locale fr
```

Make sure that your website is now translated in French at http://localhost:3000/fr/.

Automate with CI

We will configure the CI to **download the Crowdin translations at build time** and keep them outside of Git.

Add `website/i18n` to `.gitignore`.

Set the `CROWDIN_PERSONAL_TOKEN` env variable on your CI.

Create an npm script to `sync` Crowdin (extract sources, upload sources, download translations):

package.json

```json
{
  "scripts": {
    "crowdin:sync": "docusaurus write-translations && crowdin upload && crowdin download"
  }
}
```

Call the `npm run crowdin:sync` script in your CI, just before building the Docusaurus site.

TIP

Keep your deploy-previews fast: don't download translations, and use `npm run build -- --locale en` for feature branches.

CAUTION

Crowdin does not support well multiple concurrent uploads/downloads: it is preferable to only include translations to your production deployment, and keep deploy previews untranslated.

Advanced Crowdin topics

MDX

CAUTION

Pay special attention to the JSX fragments in MDX documents!

Crowdin **does not support officially MDX**, but they added **support for the `.mdx` extension**, and interpret such files as Markdown (instead of plain text).

MDX problems

Crowdin thinks that the JSX syntax is embedded HTML and can mess up with the JSX markup when you download the translations, leading to a site that fails to build due to invalid JSX.

Simple JSX fragments using simple string props like `<Username name="Sebastien"/>` will work fine; more complex JSX fragments using object/array props like `<User person={{name: "Sebastien"}}/>` are more likely to fail due to a syntax that does not look like HTML.

MDX solutions

We recommend extracting the complex embedded JSX code as separate standalone components. We also added an `mdx-code-block` escape hatch syntax:

~~~
# How to deploy Docusaurus

To deploy Docusaurus, run the following command:

````mdx-code-block
import Tabs from '@theme/Tabs';

import TabItem from '@theme/TabItem';

<Tabs>
  <TabItem value="bash" label="Bash">

  ```bash
  GIT_USER=<GITHUB_USERNAME> yarn deploy
  ```

  </TabItem>
  <TabItem value="windows" label="Windows">

  ```batch
  cmd /C "set "GIT_USER=<GITHUB_USERNAME>" && yarn deploy"
  ```

  </TabItem>
</Tabs>
````
~~~

This will:

- be interpreted by Crowdin as code blocks (and not mess-up with the markup on download)

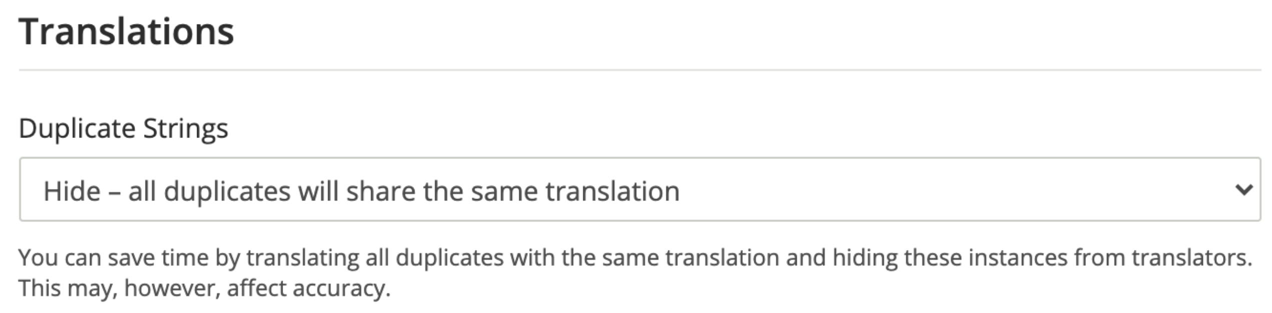
- be interpreted by Docusaurus as regular JSX (as if it was not wrapped by any code block)
- unfortunately opt-out of MDX tooling (IDE syntax highlighting, Prettier...)

Docs versioning

Configure translation files for the `website/versioned_docs` folder.

When creating a new version, the source strings will generally be quite similar to the current version (`website/docs`), and you don't want to translate the new version docs again and again.

Crowdin provides a `Duplicate Strings` setting.



Translations

Duplicate Strings

Hide – all duplicates will share the same translation

You can save time by translating all duplicates with the same translation and hiding these instances from translators. This may, however, affect accuracy.

We recommend using `Hide`, but the ideal setting depends on how much your versions are different.

⚠ CAUTION

Not using `Hide` leads to a much larger amount of `source strings` in quotas, and will affect the pricing.

Multi-instance plugins

You need to configure translation files for each plugin instance.

If you have a docs plugin instance with `id=ios`, you will need to configure those source files as well

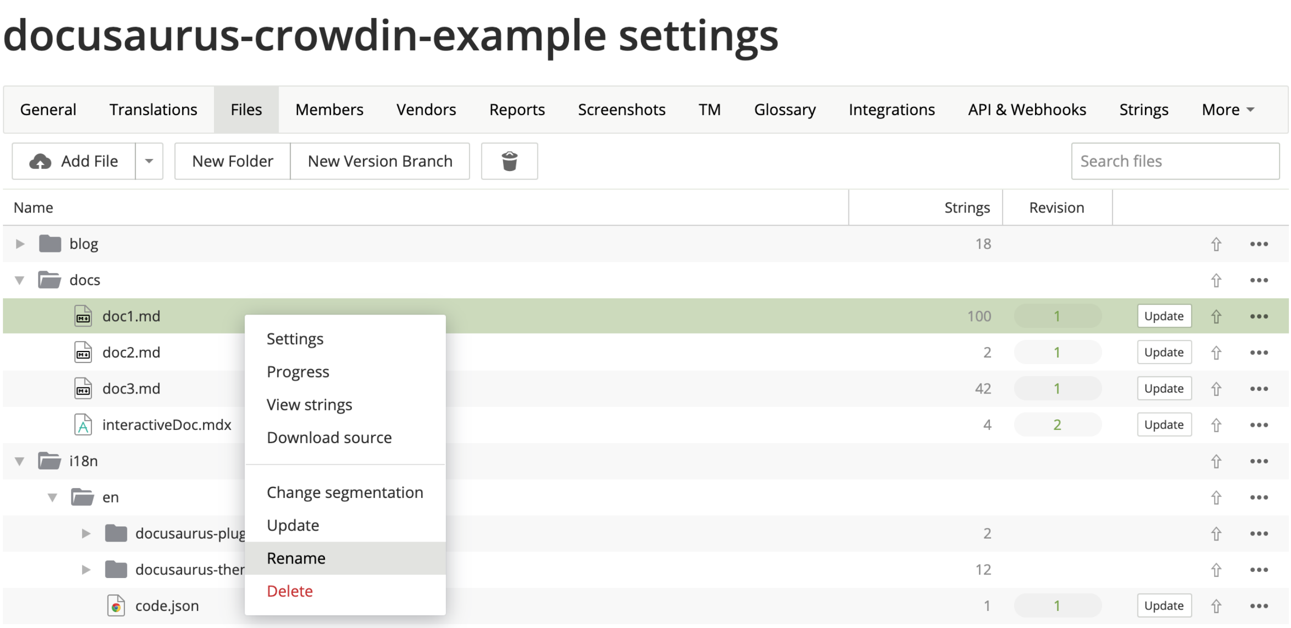
- `website/ios`
- `website/ios_versioned_docs` (if versioned)

Maintaining your site

Sometimes, you will **remove or rename a source file** on Git, and Crowdin will display CLI warnings:

```
2:53:51 PM: ✔  Building ZIP archive with the latest translations
2:53:51 PM: [●••••] Fetching project info
2:53:51 PM: [•●•••] Fetching project info
2:53:51 PM: ✔  Building translation (100%)
2:53:51 PM: [●••••] Downloading translation
2:53:51 PM: ✔  Downloading translation
2:53:52 PM: ✔  Extracted: 'i18n/fr/code.json'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-blog/2019-05-28-hola.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-blog/2019-05-29-hello-world.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-blog/2019-05-30-welcome.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-docs/current.json'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-docs/current/doc1.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-docs/current/doc2.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-docs/current/doc3.md'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-plugin-content-docs/current/interactiveDoc.mdx'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-theme-classic/footer.json'
2:53:52 PM: ✔  Extracted: 'i18n/fr/docusaurus-theme-classic/navbar.json'
2:53:52 PM: ⚠  Downloaded translations don't match the current project configuration. The translations for the following sources
will be omitted (use --verbose to get the list of the omitted translations):
        - docs/doc4.md (1)
2:53:52 PM: Done in 20.84s.
```

When your sources are refactored, you should use the Crowdin UI to **update your Crowdin files manually**:



docusaurus-crowdin-example settings

Name	Strings	Revision
blog	18	
docs		
doc1.md	100	1
doc2.md	2	1
doc3.md	42	1
interactiveDoc.mdx	4	2
i18n		
en		
docusaurus-plug	2	
docusaurus-ther	12	
code.json	1	1

VCS (Git) integrations

Crowdin has multiple VCS integrations for GitHub, GitLab, Bitbucket.

🔥 TL;DR

We recommend avoiding them.

It could have been helpful to be able to edit the translations in both Git and Crowdin, and have a **bi-directional sync** between the 2 systems.

In practice, **it didn't work very reliably** for a few reasons:

- The Crowdin -> Git sync works fine (with a pull request)
- The Git -> Crowdin sync is manual (you have to press a button)
- The heuristics used by Crowdin to match existing Markdown translations to existing Markdown sources are not 100% reliable, and you have to verify the result on Crowdin UI after any sync from Git
- 2 users concurrently editing on Git and Crowdin can lead to a translation loss
- It requires the `crowdin.yml` file to be at the root of the repository

In-Context localization

Crowdin has an In-Context localization feature.

⚠ CAUTION

Unfortunately, it does not work yet for technical reasons, but we have good hope it can be solved.

Crowdin replaces Markdown strings with technical IDs such as `crowdin:id12345`, but it does so too aggressively, including hidden strings, and messes up with front matter, admonitions, JSX...

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`.

You may prefer the edit button to link to the Crowdin interface instead by using the `editUrl` function to customize the edit URLs on a per-locale basis.

docusaurus.config.js

```
const DefaultLocale = 'en';

module.exports = {
  presets: [
    [
      '@docusaurus/preset-classic',
      {
        docs: {
          editUrl: ({locale, versionDocsDirPath, docPath}) => {
            // Link to Crowdin for French docs
            if (locale !== DefaultLocale) {
              return `https://crowdin.com/project/docusaurus-v2/${locale}`;
            }
            // Link to GitHub for English docs
            return `https://github.com/facebook/docusaurus/edit/main/website/${versionDocsDirPath}/${docPath}
          },
        },
        blog: {
          editUrl: ({locale, blogDirPath, blogPath}) => {
            if (locale !== DefaultLocale) {
              return `https://crowdin.com/project/docusaurus-v2/${locale}`;
            }
            return `https://github.com/facebook/docusaurus/edit/main/website/${blogDirPath}/${blogPath}`;
          },
        },
      },
    ],
  ],
};
```

ⓘ NOTE

It is currently **not possible to link to a specific file** in Crowdin.

Example configuration

The **Docusaurus v2 configuration file** is a good example of using versioning and multi-instance:

crowdin.yml

```yaml
project_id: '428890'
api_token_env: CROWDIN_PERSONAL_TOKEN
preserve_hierarchy: true
languages_mapping: &languages_mapping
  two_letters_code:
    pt-BR: pt-BR
files:
  - source: /website/i18n/en/**/*
    translation: /website/i18n/%two_letters_code%/**/%original_file_name%
    languages_mapping: *languages_mapping
  - source: /website/docs/**/*
    translation: /website/i18n/%two_letters_code%/docusaurus-plugin-content-docs/current/**/%original_file_na
    languages_mapping: *languages_mapping
  - source: /website/community/**/*
    translation: /website/i18n/%two_letters_code%/docusaurus-plugin-content-docs-community/current/**/%origir
    languages_mapping: *languages_mapping
  - source: /website/versioned_docs/**/*
    translation: /website/i18n/%two_letters_code%/docusaurus-plugin-content-docs/**/%original_file_name%
    languages_mapping: *languages_mapping
  - source: /website/blog/**/*
    translation: /website/i18n/%two_letters_code%/docusaurus-plugin-content-blog/**/%original_file_name%
    languages_mapping: *languages_mapping
  - source: /website/src/pages/**/*
    translation: /website/i18n/%two_letters_code%/docusaurus-plugin-content-pages/**/%original_file_name%
    ignore: [/**/*.js, /**/*.jsx, /**/*.ts, /**/*.tsx, /**/*.css]
    languages_mapping: *languages_mapping
```

Edit this page

*Last updated on **Mar 23, 2023** by **Sébastien Lorber***

Previous
« i18n - Using git

Next
What's next? »

Learn

Introduction

Installation

Migration from v1 to v2

Community

Stack Overflow

Feature Requests

Discord

Help

More

Blog

Changelog

GitHub

Twitter

Legal

Privacy

Terms

Data Policy

Cookie Policy