# Contributing to Uppload

?? First off, thanks for taking the time to contribute! ??

The following is a set of guidelines for contributing to Uppload and its packages. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

## Code of Conduct

This project and everyone participating in it is governed by the [Uppload Code of Conduct](https://github.com/elninotech/uppload/blob/master/CODE\_OF\_CONDUCT.md). By participating, you are expected to uphold this code. Please report unacceptable behavior to [info@elnino.tech](mailto:info@elnino.tech).

## Gitmoji

We use [Gitmoji](https://gitmoji.carloscuesta.me) when writing commits to automatically generate changelogs, releases, etc. If you're writing code for Uppload, we highly encourage you to use Gitmoji.

## Icons

When adding a new service or effect, make sure you design an SVG icon using the following settings. This way, we can theme it using the CSS `fill` property:

- 256px square

- Black fill color (#000)

- Convert all text to outlines

- No borders, only shapes

Then, make sure you compress the icon using [SVGOMG](https://jakearchibald.github.io/svgomg/). On average, we've seen that a ~800 byte SVG icon can be compressed to ~250 bytes. Use the minimum precision setting that does not distort the icon and remove all metadata except for xmlns.

Finally, remove the `width` and `height` attributes from the SVG and use `viewbox` to allow scaling, and add the `aria-hidden="true"` attribute to help screen readers ignore the icon. Then, the beginning of your SVG string should be `<svg aria-hidden="true" viewBox="0 0 256 256" xmlns="http://www.w3.org/2000/svg">`.

## Translations

To edit a language translation, head to the file in the [`./src/i18n`](https://github.com/elninotech/uppload/tree/master/src/i18n) folder and click on the Edit (pencil) icon on the top-right of the GitHub file contents UI. This will create a fork of the project and you can make a pull request with the changes. Alternately, you can also edit it in your preferred code editor after forking the project.

If you want to add a new language, create a new TypeScript file with the two-letter ISO 639-2 code in the [`./src/i18n`](https://github.com/elninotech/uppload/tree/master/src/i18n) folder that exports an object containing the key-value pairs, named the language code. For example, if you're adding a Dutch translation, create an `nl.ts` file that starts with: `export const nl = {};`. You can use the [`en.ts`](https://github.com/elninotech/uppload/blob/master/src/i18n/en.ts) as a template.

In both cases, since we use Gitmoji, your commit message should be "\:globe\_with\_meridians: \[Update or Add] translation for \[Language name]".

### Variables

We use [i18n helpers](https://github.com/elninotech/uppload/blob/master/src/helpers/i18n.ts) to find and replace variables in translations. [I18N documentation](https://uppload.js.org/i18n)

For example, in the string "Import from $1$", we replace $1$ with "Instagram" to generate "Import from Instagram". Similarly, the n-th variable replaces $n$ ($1$, $2$, $3$, and so on). You can also define a custom language-specific helper function, which is applied before returning the translation. For example, [`en.ts`](https://github.com/elninotech/uppload/blob/master/src/i18n/en.ts) replaces $A$ with "a" or "an" depending on whether the next word starts with a vowel. In this example, "Enter $A$ $1$ $2$ URL" may be compiled to "Enter an Instagram file URL".

## Online one-click setup

You can use Gitpod (a free online VS Code-like IDE) for contributing online. With a single click, it will launch a workspace using which you make a Pull Request from within the browser. Gitpod will automatically:

- Clone the repository

- Install all required dependencies

- Run the local development script: `npm run demo`

[![Open in Gitpod](https://gitpod.io/button/open-in-gitpod.svg)](https://gitpod.io/from-referrer/)

## Build & deploy

When working locally, the `yarn demo` script is used for a local development setup with hot reloading.

When deploying a new version of Uppload, follow these steps:

1. Increment `package.json` version based on [SemVer](https://semver.org/)

2. Update `CHANGELOG.md` using `yarn changelog`

3. Build a new package using `yarn build`

4. Publish the newly built package using `unpm publish`

5. Create a new release on GitHub with version name (e.g., `v.2.1.1`)