# Contributing

Suggestions and pull requests are highly encouraged! Have a look at the [open issues](https://github.com/sindresorhus/refined-github/issues?q=is%3Aissue+is%3Aopen+label%3A%22help+wanted%22+sort%3Areactions-%2B1-desc), especially [the easy ones](https://github.com/sindresorhus/refined-github/issues?q=is%3Aissue+is%3Aopen+label%3A%22good+first+issue%22+sort%3Areactions-%2B1-desc).

## Notions

- You will need to be familiar with [npm](https://docs.npmjs.com/getting-started/) and TypeScript to build this extension.

- The extension can be loaded into Chrome or Firefox manually ([See notes below](#loading-into-the-browser))

- [JSX](https://reactjs.org/docs/introducing-jsx.html) is used to create DOM elements.

- All the [latest DOM APIs](https://github.com/WebReflection/dom4#features) and JavaScript features are available because the extension only has to work in the latest Chrome and Firefox. ?

- Each JavaScript feature lives in its own file under [`source/features`](https://github.com/sindresorhus/refined-github/tree/master/source/features) and it's imported in [`source/refined-github.ts`](https://github.com/sindresorhus/refined-github/blob/master/source/refined-github.ts).

- See what a \_feature\_ [looks like](https://github.com/sindresorhus/refined-github/blob/master/source/features/user-profile-follower-badge.tsx).

- Follow [the styleguide](https://github.com/sindresorhus/refined-github/blob/master/readme.md#L100) that appears in the Readme's source to write readable descriptions.

- Refined GitHub tries to integrate as best as possible, so [GitHub's own styleguide](https://primer.style/css) might come in useful.

## `features.add`

The simplest usage of `feature.add` is the following. This will be run instantly on all page-loads (but not on ajax loads):

```js

import \* as pageDetect from 'github-url-detection';

import features from '.';

function init () {

console.log('?');

}

features.add({

id: \_\_filebasename,

description: 'Simplify the GitHub interface and adds useful features',

screenshot: 'https://user-images.githubusercontent.com/1402241/58238638-3cbcd080-7d7a-11e9-80f6-be6c0520cfed.jpg',

}, {

include: [

pageDetect.isPR

],

waitForDomReady: false,

init

});

```

Here's an example using all of the possible `feature.add` options:

```ts

import React from 'dom-chef';

import select from 'select-dom';

import delegate from 'delegate-it';

import \* as pageDetect from 'github-url-detection';

import features from '.';

function append(event: delegate.Event<MouseEvent, HTMLButtonElement>): void {

event.delegateTarget.after('?', <div className="rgh-jsx-element">Button clicked!</div>);

}

function init(): void {

// Events must be set via delegate, unless shortlived

delegate(document, '.btn', 'click', append);

}

features.add({

id: \_\_filebasename,

description: 'Simplify the GitHub interface and adds useful features',

screenshot: 'https://user-images.githubusercontent.com/1402241/58238638-3cbcd080-7d7a-11e9-80f6-be6c0520cfed.jpg',

shortcuts: { // This only adds the shortcut to the help screen, it doesn't enable it

'?': 'Edit your last comment'

},

}, {

/\*\* Whether to wait for DOM ready before runnin `init`. `false` makes `init` run right as soon as `body` is found. @default true \*/

waitForDomReady: false,

/\*\* Whether to re-run `init` on pages loaded via AJAX. @default true \*/

repeatOnAjax: false,

/\*\* Rarely needed: When pressing the back button, the DOM and listeners are still there, so normally `init` isn’t called again. If this is true, it’s called anyway. @default false \*/

repeatOnAjaxEvenOnBackButton: true,

include: [

pageDetect.isUserProfile,

pageDetect.isRepo

],

exclude: [

pageDetect.isOwnUserProfile

],

init

}, {

include: [

pageDetect.isGist

],

init: () => console.log('Additional listener for gist pages!')

});

```

## Requirements

[Node.js](https://nodejs.org/en/download/) version 13 or later is required.

## Workflow

First clone:

```sh

git clone https://github.com/sindresorhus/refined-github

cd refined-github

npm install

```

When working on the extension or checking out branches, use this to have it constantly build your changes:

```sh

npm run watch # Listen to file changes and automatically rebuild

```

Then load or reload it into the browser to see the changes.

## Loading into the browser

Once built, load it in the browser of your choice.

```sh

npm run start # Open extension in Chrome

```

```sh

npm run start:firefox # Open extension in Firefox

```

Or you can [load it manually in Chrome](https://www.smashingmagazine.com/2017/04/browser-extension-edge-chrome-firefox-opera-brave-vivaldi/#google-chrome-opera-vivaldi) or [Firefox](https://www.smashingmagazine.com/2017/04/browser-extension-edge-chrome-firefox-opera-brave-vivaldi/#mozilla-firefox).