# Contributing to Augury

:sparkling\_heart: First things first, thank you for contributing. :sparkling\_heart: We know

it takes a lot of time and effort. Everything in this document is meant

to provide you with as much insight as possible with regards to our

contribution standards. As you read it, it is important to remember that

these are just \_\_guidelines\_\_ and not rules. Please use your best

judgement when applying them.

## Table of Contents

1. [I'm new, anything I should know?](#im-new-anything-i-should-know)

1. [Behaving yourself](#behaving-yourself)

1. [Environmnet setup](#environment-setup)

1. [Contributing workflow](#contributing-workflow)

1. [Making changes](#making-changes)

1. [How do I get in touch?](#how-do-i-get-in-touch)

1. [License](#license)

## I'm new, anything I should know?

:wave: Hello and welcome. :wave: The first thing you should do is \_read\_ the [README](./README.md).

We've put a lot of love and care in writing it and it should provide you with the context you need

for contributing to this project. Really, you should read it.

The next thing you should do is take a look at the [Architecture of this extension](./docs/ARCHITECTURE.md).

## Behaving yourself

In the interest of making the Augury project a safe and friendly

place for people from diverse backgrounds, we'll be adhering to a [Contributor Code of Conduct](./CODE\_OF\_CONDUCT.md). As a contributor, you'll be expected to

uphold this code as well as report unacceptable behaviour to

[augury@rangle.io](mailto:augury@rangle.io).

## Environment setup

### Download

1. Fork the repo and create your branch from master.

1. Add rangle/augury to your remotes git remote add upstream git@github.com:rangle/augury.git.

1. After cloning to local directory, create your own branch.

### Using Git

Here at Rangle we strive to adhere to the following guidelines. It is expected that any PR that is made will be branched off a forked rangle/augury master and have a sensible name with a "feature/bug/chore" prefix (e.g. feature-colour-picker).

Run `git fetch upstream` - if there are changes to rangle/augury.

Run `git rebase upstream/master` - to fast forward your local branch to upstream.

Don't forget to squash your commits. [git-scm.com](https://git-scm.com/book/en/v2/Git-Tools-Rewriting-History)

### Building and testing

1. Run `npm install`.

1. Run `npm run build` (errors related to typing files conflicts can be ignored for now).

1. Navigate to:

- For Chrome: `chrome://extensions` and enable Developer Mode

- For Firefox: `about:debugging#addons`

1. Choose "Load unpacked extension" (Chrome) or "Load Temporary Add-On" (Firefox).

1. In the dialog, open your augury directory.

You can test out the extension with a quick example app. In a different terminal window:

1. Git clone the https://github.com/rangle/augury-examples project.

1. Navigate to `kitchen-sink-demo` directory.

1. Run `npm install`.

1. Run `npm start` on the kitchen-sink-example root folder.

### Coding style

\* Use semicolons;

\* Commas last,

\* 2 spaces for indentation (no tabs)

\* Prefer `'` over `"`

\* 120 character line length

\* ...

### Recommendations

\* Augury `npm start` script sets up webpack to run with a watcher on your files, so when you make changes, you don't have to rebuild the extension (it'll do it for you).

\* If the app becomes unresponsive, or a change is not visible, you might need to close and reopen the dev tools.

\* Failing the above, you might have to go to `chrome://extensions` (Chrome) or `about:debugging#addons` (Firefox) and click on the Reload button.

\* You can ignore some warnings for now.

## Contributing workflow

Remember, no contribution is too small and no issue is too trivial. If you see a typo, we'd love to get a PR. We actually hate typos...they're the worst.

### Bugs

:beetle: We are using [GitHub Issues](https://github.com/rangle/augury/issues) for our public bugs. We will keep track on this and will to make it clear when we have an internal fix in progress. Before filing for a new task, try to make sure your problem doesn't already exist.

If you do find a new issue, the best way to get a bug fixed is to provide a test case with either one of the example apps bundled in the repo, or by making your own Angular application illustrating the reduced use case. Don't forget to include the version of Angular app you're using!

At the moment, we offer limited backwards compatibility until Angular stabilizes. Please check [README](./README.md#supported-version) for the version Augury currently supports.

### Pull requests

:shipit: The core team will be monitoring for pull requests.

\*Before\* submitting a pull request, please make sure the following is done...

1. Fork the repo and create your branch from `dev`.

2. If you've added code that should be tested, add tests.

3. If you've changed APIs, development workflow or aspects of the architecture, update the corresponding documentation.

4. TypeScript linter is part of our build task, make sure your code lints before submitting your PRs.

More information can be found in `tslint.json` file of this repository.

TypeScript linter is part of our build task and will run in either `npm run build` or `npm start`

## Making changes

So you want to make some changes to the application. Again, we recommend taking a look at the [Augury Architecture](./docs/ARCHITECTURE.md) before trying to make changes to the app, as it will give you a better idea of where to find things.

### Front-End

So you want to develop on the front-end (make some logic or visual changes to the extension). This is a self contained Angular app running as an extension, so getting acquainted with [Angular](https://angular.io/) is a good idea. We are using a [Flux](https://facebook.github.io/flux/) style application architecture with the help of [RxJS](https://github.com/Reactive-Extensions/RxJS).

\_frontend structure overview\_

\* Actions

\* example-actions

\* Components

\* example-component

\* example-controller

\* example-template

\* Stores

\* example-store

#### Components

The extension is separated into various components, which can be found in batarangle/frontend/components. If you want to make changes to specific parts of the application, you have to figure out what component they belong to.

#### Controllers

Each of these components will have their .ts file containing the component class and Angular decorators (like @Component or @View). If you want to modify any logic that deals with interaction with the component, this is most likely where you would find it. Most of the logic, constructor, dependencies and related actions are placed within the component class. Any necessary actions, stores or other classes can be imported into the file.

#### Templates

If you want to modify the front-end extension or the application's styles, this is where you would do it. If the inline HTML from @View in the controller is too large (or simply due to preference), you can move the templates to their own file within the component.

#### Actions

Flux defines an action as a method exposed by the dispatcher that allows us to trigger a dispatch to the stores, and to include a payload of data. There is a place where application actions are defined in various lists, due to separation of concerns (e.g. user and backend actions). These are meant to be imported into controllers to take care of any actions that specific component might want to accomplish (especially those using the Rx dispatcher).

#### Stores

\*Stores contain the application state and logic.\*

They are meant to manage the application state, as opposed to simply keeping a record of data, or managing collections of data. Similar to the model in traditional MVC, stores manage the state for a particular domain within the application, meaning they are associated with keeping, storing, and sending of that particular domain's data.

## How do I get in touch?

We are using Slack for all of our communication. If you want to contribute or need help getting started, join us on [Slack](https://augury-slack.herokuapp.com/).

## License

[MIT](LICENSE)