# Contributing to Flight

Looking to contribute something to Flight? Here's how you can help.

## Bugs reports

A bug is a \_demonstrable problem\_ that is caused by the code in the

repository. Good bug reports are extremely helpful – thank you!

Guidelines for bug reports:

1. \*\*Use the GitHub issue search\*\* &mdash; check if the issue has already been

reported.

2. \*\*Check if the issue has been fixed\*\* &mdash; try to reproduce it using the

latest `master` or development branch in the repository.

3. \*\*Isolate the problem\*\* &mdash; ideally create a reduced test

case and a live example.

4. Please try to be as detailed as possible in your report. Include specific

information about the environment – operating system and version, browser

and version, version of Flight – and steps required to reproduce the issue.

## Feature requests & contribution enquiries

Feature requests are welcome. But take a moment to find out whether your idea

fits with the scope and aims of the project. It's up to \*you\* to make a strong

case for the inclusion of your feature. Please provide as much detail and

context as possible.

Contribution enquiries should take place before any significant pull request,

otherwise you risk spending a lot of time working on something that we might

have good reasons for rejecting.

## Pull requests

Good pull requests&mdash;patches, improvements, new features&mdash;are a fantastic

help. They should remain focused in scope and avoid containing unrelated

commits.

Make sure to adhere to the coding conventions used throughout the codebase

(indentation, accurate comments, etc.). Please run `npm test` before you push:

this will run the jasmine test suite (using karma runner) and check your syntax

style is consistent with master (using JSCS)

Please follow this process; it's the best way to get your work included in the

project:

1. [Fork](http://help.github.com/fork-a-repo/) the project, clone your fork,

and configure the remotes:

```bash

# Clone your fork of the repo into the current directory

git clone https://github.com/<your-username>/flight

# Navigate to the newly cloned directory

cd <repo-name>

# Assign the original repo to a remote called "upstream"

git remote add upstream git://github.com/twitter/flight

```

2. If you cloned a while ago, get the latest changes from upstream:

```bash

git checkout master

git pull upstream master

```

3. Install the dependencies (you must have Node.js and [Bower](http://bower.io)

installed), and create a new topic branch (off the main project development

branch) to contain your feature, change, or fix:

```bash

npm install

bower install

git checkout -b <topic-branch-name>

```

4. Make sure to update, or add to the tests when appropriate. Patches and

features will not be accepted without tests. Run `make test` to check that

all tests pass after you've made changes.

5. Commit your changes in logical chunks. Provide clear and explanatory commit

messages. Use Git's [interactive rebase](https://help.github.com/articles/interactive-rebase) feature to tidy up

your commits before making them public.

6. Locally merge (or rebase) the upstream development branch into your topic branch:

```bash

git pull [--rebase] upstream master

```

7. Push your topic branch up to your fork:

```bash

git push origin <topic-branch-name>

```

8. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/)

with a clear title and description.

9. If you are asked to amend your changes before they can be merged in, please

use `git commit --amend` (or rebasing for multi-commit Pull Requests) and

force push to your remote feature branch. You may also be asked to squash

commits.

## License

By contributing your code,

You agree to license your contribution under the terms of the MIT License

https://github.com/twitter/flight/blob/master/LICENSE