Marionette has a few guidelines to facilitate your contribution and streamline

the process of getting changes merged in and released.

1. [Setting up Marionette locally](#setting-up-marionette-locally)

2. [Reporting a bug](#reporting-a-bug)

3. [Submitting patches and fixes](#submitting-patches-and-fixes)

4. [Running Tests](#running-tests)

## Setting up Marionette locally

\* Fork the Marionette repo.

\* `git clone` your fork onto your computer.

\* Run `yarn install` to make sure you have all Marionette dependencies locally.

\* Run `yarn build` to build source files.

## Reporting a bug

In order to best help out with bugs, we need to know the following information

in your bug submission:

\* Marionette version #.

\* Backbone version #.

Including this information in a submission will help us test the problem and

ensure that the bug is both reproduced and corrected on the platforms /

versions that you are having issues with.

<a name="format-desc"></a>\*\*Provide A Meaningful Description\*\*

It is very important to provide a meaningful description with your bug reports

and pull requests. A good format for these descriptions will include the

following things:

1. The problem you are facing (in as much detail as is necessary to describe

the problem to someone who doesn't know anything about the system you're

building)

2. A summary of the proposed solution

3. A description of how this solution solves the problem, in more detail than

item #2

4. Any additional discussion on possible problems this might introduce,

questions that you have related to the changes, etc.

For a PR, we need at least the first 2 items to understand why you are changing

the code. If not, we will ask that you add the necessary information.

Please refrain from giving code examples in altJS languages like CoffeeScript,

etc. Marionette is written in plain-old JavaScript and is generally easier for all

members in the community to read.

### When you don't have a bug fix

If you are stuck in a scenario that fails in your app, but you don't know how to

fix it, submit a failing spec to show the failing scenario. Follow the

guidelines for a pull request submission, but don't worry about fixing the

problem. A failing spec to show that a problem exists is a very very very

helpful pull request for us.

We'll even accept a failing test pasted into the ticket description instead of a

PR. That would at least get us started on creating the failing test in the code.

## Submitting patches and fixes

See [Github's documentation for pull

requests](https://help.github.com/articles/using-pull-requests).

Pull requests are by far the best way to contribute to Marionette. They are by

far the easiest way to demonstrate issues and your proposed resolution. To

really help us evaluate your pull request and bring it into Marionette, please

provide as much information as possible and follow the guidelines below:

1. Determine the branch as your base: `next` or `master`

2. Provide a brief summary of what your pull request is doing

3. Reference any relevant Github issue numbers

4. Include any extra detail you feel will help provide context

### Determining your branch

When submitting your pull request, you need to determine whether to base off

`next` or `master`:

\* If you're submitting a bug fix, base off `next`

\* If you're submitting a new feature, base off `next`

\* If you're submitting documentation for a new feature, base off `next`

\* If you're submitting documentation for the current release, base off `master`

### Submitting a Great Patch

We want Marionette to provide a great experience to developers and help you

write great applications using it. To help us achieve this goal, please follow

these guidelines when submitting your patches.

#### Solving Issues

When you're submitting a bug fix, include spec tests, where applicable, showing

the issue and the resolution. We strive to maintain 100% code coverage in our

testing.

#### Coding Guidelines

The Marionette coding conventions are provided in the ESLint configuration

included in the repository. Most IDEs and text editors will provide, or allow

for, a plugin for ESLint to read the `.eslintrc` file.

For areas where the configuration provides no guidance, try to stick to the

conventions in the file you're editing.

#### How we Approve Pull Requests

We utilise Github's review approach. When receiving your pull request, we will

comment inline and provide guidance to help you get your pull request merged

into Marionette. This is not a one-way process and we're more than happy to

discuss the context of your decisions.

Once two Marionette.js members approve the pull request, we will then merge it

into the base branch.

Please remember that Marionette is a community-maintained project and, as such,

many of us are working on this in our spare time. If we haven't commented on

your pull request, please be patient. We may be available on our Gitter channel

to discuss further.

## Running Tests

\* via command-line by running `yarn test`

\* in the browser by running `yarn test-browser`

To see the test matrix - run `yarn coverage`

## Writing Tests and Code Style

[More information]('test/unit/README.md')