# How to contribute

## Issue Submission Guidelines

If you are trying to get a particular effect or you have a problem with your code, please ask your question [on stackoverflow.com](http://stackoverflow.com/questions/tagged/dc.js) or the [user group](https://groups.google.com/forum/?fromgroups#!forum/dc-js-user-group). We use the issue tracker for bug reports and feature requests only.

Get help faster with a working example! Fork these to get started:<br>

[blank jsFiddle](https://jsfiddle.net/gordonwoodhull/rL82bguk/1) - [example jsFiddle](https://jsfiddle.net/gordonwoodhull/5ztavmjy/2) - [blank bl.ock](https://blockbuilder.org/gordonwoodhull/f6bab3d2f5b34018548207014b4056bf) - [example bl.ock](https://blockbuilder.org/gordonwoodhull/bcf9eaa0bfc2c84373cffac06d5755e5)

For bug reports and feature requests submit a [github issue](http://github.com/dc-js/dc.js/issues)

\* Please include the version of DC you are using

\* If you can, please try the latest version of DC on the [master](https://raw.github.com/dc-js/dc.js/master/dc.js) branch to see if your issue has already been addressed or is otherwise affected by recent changes.

## Development Guidelines

\* Code is written and distributed as ES6 modules.

\* `src/` folder is directly exposed for ES6 module users.

In addition, a `rollup` generated UMD bundle is placed in `dist/`.

\* Please try to follow the existing code formatting.

\* Make changes to the files in `src/` not `dc.js`.

\* Add tests to `spec/`. Feel free to create a new file if needed.

`spec/index.html` is a generated file, no need to manually update.

\* Please add adequate tests. Many good PRs are not merged for the

lack of test cases.

\* Please use `.transitionDuration(0)` for all chart tests.

\* If you have made significant changes or added a new chart type,

please consider adding a new example in `web-src/examples/`.

\* Source for `dc.css` is maintained in `style/dc.scss`.

\* All files in `web/` and `dist/` are generated.

Do not make direct changes to these.

\* Run `grunt test` to run all tests using Karma and Jasmine on headless Firefox,

or see [Testing](#Testing) for other methods

\* We use eslint to check coding guidelines and conventions.

Run `grunt lint` to confirm that your code meets the dc.js style guidelines.

See `.eslintrc` for current rules.

\* The default grunt task will install a git pre-commit hook

`.git/hooks/pre-commit` to help verify the coding conventions.

Run `grunt` without any arguments to install the hook.

You only need to do it once.

\* Commit your changes to `src/\*`, `spec/\*`, and `web-src/\*`; exclude build artifacts.

(Build artifacts include `dist/\*` and `web/\*`).

## Testing

\* `grunt test` will run will all tests using Karma and Jasmine on headless Firefox.

`grunt ci` tests headless Chrome as well.

Some IDEs facilitate executing tests from the IDE itself.

\* Running `grunt server` starts a server at http://localhost:8888/ which serves

the current directory as root.

\* Tests can also be run on the browser by going to http://localhost:8888/spec when

server is running.

\* With the server running, go to http://localhost:8888/web to check the stocks example.

You can also go to the examples sub folder to test specific examples.

\* During development, you can run `grunt test-n-serve`.

It will start the server and in addition keep a watch for changes to

run build/test as needed.

\* If your changes might affect animated transitions,

go to the relevant transition tests in

http://localhost:8888/web/transitions and watch them by eye to see if they make

sense, or add another one if needed. There are no automated tests for transitions.

## Pull Request Guidelines

\* Fork the repository.

\* As with all pull requests, put your changes in a branch.

For contributions that change the dc.js API, create your branch off of `develop`.

If your contribution does not change the API, branch off of `master` instead.

\* Make your changes.

\* Submit a pull request.

\* Travis CI is linked to PRs, so, all tests will get executed on creation/updation of a PR.

The status is visible on the GitHub PR page.

\* If you need to merge `develop` or `master` into your patchset,

instead of merging please use git rebase.

\* The DC maintainer team will review and build the artifacts when merging.

\* If you continue making changes to your fork of `dc.js`,

please create a separate branch for each pull request and keep the changes separate.