 [Chart.js](http://docs.google.com/docs/3.9.1/)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

* [Chart.js](http://docs.google.com/docs/3.9.1/)
* Getting Started
* General
* Configuration
* Chart Types
* Axes
* Developers
  + [Developers](http://docs.google.com/docs/3.9.1/developers/)
  + [API](http://docs.google.com/docs/3.9.1/developers/api.html)
  + [New Axes](http://docs.google.com/docs/3.9.1/developers/axes.html)
  + [New Charts](http://docs.google.com/docs/3.9.1/developers/charts.html)
  + [Contributing](http://docs.google.com/docs/3.9.1/developers/contributing.html)
  + [Plugins](http://docs.google.com/docs/3.9.1/developers/plugins.html)
  + [Publishing an extension](http://docs.google.com/docs/3.9.1/developers/publishing.html)
  + [TypeDoc](http://docs.google.com/docs/3.9.1/api/)
  + [Updating Charts](http://docs.google.com/docs/3.9.1/developers/updates.html)

[**#**](#gjdgxs) Contributing

New contributions to the library are welcome, but we ask that you please follow these guidelines:

* Before opening a PR for major additions or changes, please discuss the expected API and/or implementation by [filing an issue (opens new window)](https://github.com/chartjs/Chart.js/issues) or asking about it in the [Chart.js Slack (opens new window)](https://chartjs-slack.herokuapp.com/) #dev channel. This will save you development time by getting feedback upfront and make review faster by giving the maintainers more context and details.
* Consider whether your changes are useful for all users, or if creating a Chart.js [plugin](http://docs.google.com/docs/3.9.1/developers/plugins.html) would be more appropriate.
* Check that your code will pass tests and eslint code standards. npm test will run both the linter and tests for you.
* Add unit tests and document new functionality (in the test/ and docs/ directories respectively).
* Avoid breaking changes unless there is an upcoming major release, which is infrequent. We encourage people to write plugins for most new advanced features, and care a lot about backwards compatibility.
* We strongly prefer new methods to be added as private whenever possible. A method can be made private either by making a top-level function outside of a class or by prefixing it with \_ and adding @private JSDoc if inside a class. Public APIs take considerable time to review and become locked once implemented as we have limited ability to change them without breaking backwards compatibility. Private APIs allow the flexibility to address unforeseen cases.

## [**#**](#30j0zll) Joining the project

Active committers and contributors are invited to introduce yourself and request commit access to this project. We have a very active Slack community that you can join [here (opens new window)](https://chartjs-slack.herokuapp.com/). If you think you can help, we'd love to have you!

## [**#**](#1fob9te) Building and Testing

Firstly, we need to ensure development dependencies are installed. With node and npm installed, after cloning the Chart.js repo to a local directory, and navigating to that directory in the command line, we can run the following:

> npm install

This will install the local development dependencies for Chart.js.

The following commands are now available from the repository root:

> npm run build // build dist files in ./dist  
> npm run autobuild // build and watch for source changes  
> npm run dev // run tests and watch for source and test changes  
> npm run lint // perform code linting (ESLint, tsc)  
> npm test // perform code linting and run unit tests with coverage

npm run dev and npm test can be appended with a string that is used to match the spec filenames. For example: npm run dev plugins will start karma in watch mode for test/specs/\*\*/\*plugin\*.js.

### [**#**](#3znysh7) Documentation

We use [Vuepress (opens new window)](https://vuepress.vuejs.org/) to manage the docs which are contained as Markdown files in the docs directory. You can run the doc server locally using these commands:

> npm run docs:dev

### [**#**](#2et92p0) Image-Based Tests

Some display-related functionality is difficult to test via typical Jasmine units. For this reason, we introduced image-based tests ([#3988 (opens new window)](https://github.com/chartjs/Chart.js/pull/3988) and [#5777 (opens new window)](https://github.com/chartjs/Chart.js/pull/5777)) to assert that a chart is drawn pixel-for-pixel matching an expected image.

Generated charts in image-based tests should be **as minimal as possible** and focus only on the tested feature to prevent failure if another feature breaks (e.g. disable the title and legend when testing scales).

You can create a new image-based test by following the steps below:

* Create a JS file ([example (opens new window)](https://github.com/chartjs/Chart.js/blob/f7b671006a86201808402c3b6fe2054fe834fd4a/test/fixtures/controller.bubble/radius-scriptable.js)) or JSON file ([example (opens new window)](https://github.com/chartjs/Chart.js/blob/4b421a50bfa17f73ac7aa8db7d077e674dbc148d/test/fixtures/plugin.filler/fill-line-dataset.json)) that defines chart config and generation options.
* Add this file in test/fixtures/{spec.name}/{feature-name}.json.
* Add a [describe line (opens new window)](https://github.com/chartjs/Chart.js/blob/4b421a50bfa17f73ac7aa8db7d077e674dbc148d/test/specs/plugin.filler.tests.js#L10) to the beginning of test/specs/{spec.name}.tests.js if it doesn't exist yet.
* Run npm run dev.
* Click the *"Debug"* button (top/right): a test should fail with the associated canvas visible.
* Right click on the chart and *"Save image as..."* test/fixtures/{spec.name}/{feature-name}.png making sure not to activate the tooltip or any hover functionality
* Refresh the browser page (CTRL+R): test should now pass
* Verify test relevancy by changing the feature values *slightly* in the JSON file.

Tests should pass in both browsers. In general, we've hidden all text in image tests since it's quite difficult to get them passing between different browsers. As a result, it is recommended to hide all scales in image-based tests. It is also recommended to disable animations. If tests still do not pass, adjust [tolerance and/or threshold (opens new window)](https://github.com/chartjs/Chart.js/blob/1ca0ffb5d5b6c2072176fd36fa85a58c483aa434/test/jasmine.matchers.js) at the beginning of the JSON file keeping them **as low as possible**.

When a test fails, the expected and actual images are shown. If you'd like to see the images even when the tests pass, set "debug": true in the JSON file.

## [**#**](#tyjcwt) Bugs and Issues

Please report these on the GitHub page - at [github.com/chartjs/Chart.js](https://github.com/chartjs/Chart.js). Please do not use issues for support requests. For help using Chart.js, please take a look at the [chart.js (opens new window)](https://stackoverflow.com/questions/tagged/chart.js) tag on Stack Overflow.

Well structured, detailed bug reports are hugely valuable for the project.

Guidelines for reporting bugs:

* Check the issue search to see if it has already been reported
* Isolate the problem to a simple test case
* Please include a demonstration of the bug on a website such as [JS Bin (opens new window)](https://jsbin.com/), [JS Fiddle (opens new window)](https://jsfiddle.net/), or [Codepen (opens new window)](https://codepen.io/pen/). ([Template (opens new window)](https://codepen.io/pen?template=wvezeOq)). If filing a bug against master, you may reference the latest code via [https://www.chartjs.org/dist/master/chart.min.js (opens new window)](https://www.chartjs.org/dist/master/chart.min.js) (changing the filename to point at the file you need as appropriate). Do not rely on these files for production purposes as they may be removed at any time.

Please provide any additional details associated with the bug, if it's browser or screen density specific, or only happens with a certain configuration or data.

Last Updated: 8/3/2022, 12:46:38 PM

←  [New Charts](http://docs.google.com/docs/3.9.1/developers/charts.html)   [Plugins](http://docs.google.com/docs/3.9.1/developers/plugins.html)  →