

## Contributor Setup

**WARNING: We have not rechecked all these instructions after our migration to Github.**

### Setting Up Your Clone

Assuming you're using Eclipse, follow these instructions to get a Guava clone working with both Git and Maven:

1. Make sure you have the EGit and m2eclipse plugins installed in Eclipse.
2. Open the "Git Repositories" perspective. On the "Git Repositories" tab, click the "Clone a Git Repository" button.
3. Paste `https://github.com/google/guava.git` into the URI.
4. Check out all branches of the source.

You should now have a local clone of Guava on your machine, albeit it's not a Java or Maven project. We now want it to get built.

1. Run `mvn install antrun:run enforcer:enforce`, if those plugins are not already installed.
2. In Eclipse, use "File/Import", and choose "Maven/Existing Maven Projects".
3. Designate the root directory of your repository – typically `~/git/guava-libraries`.
4. Select the root `pom.xml`, labeled `guava-parent`. If you encounter Maven problems, choose "Resolve All Later."

You're almost done! You should now have several projects in Eclipse:

- `guava-parent`, the "root" project, with which you'll do all your interacting with Git: commits, pushes, pulls, rebases, checkouts, etc.
- `guava`: the main source of Guava.
- `guava-gwt`: the GWT-compatible supersource that helps make Guava GWT-compatible.
- `guava-testlib`: the test suite builders that assemble customized, exhaustive test suites for collection implementations.
- `guava-tests`: the tests for Guava.

Right-click on `guava-parent`, and go to **Team/Share Project**. Select "Git," and choose "Use repository in parent folder of project," which will tell Eclipse to use the preexisting repository that already has all the Guava history and the connection to the remote master branch.

Now you have the Guava source checked out into Eclipse, and working with Maven! Yay!

### Build and Test

*TODO: fill this in*

You'll need to use `mvn install` to build.

Pass `-Dmaven.test.skip=true` to skip tests.

## Code Reviews

Our code is formatted with `google-java-format`.

The first time you submit code to Guava, you will need to e-sign a CLA.