## 1. Use a feature branch

Feature branches make pull requests easier as GitHub will keep your pull request up-to-date when the feature branch changes. If you need to rebase to fix some things after creating your pull request, or if you end up having multiple pull requests open at the same time, \*\*having a pull request associated with a branch instead of `master` will make your job easier\*\*.

## 2. Run the test suite

\*jsonschema2pojo\* includes both unit and integration tests. The unit test coverage is not high, instead we rely on an exhaustive set of integration tests that cover all the code generation features using example schema documents.

To compile, unit test, package, and run the integration test suite (the typical full build), use:

`mvn clean verify`

## 3. Add integration tests

When contributing, you may often find unit tests (and TDD) helpful and you're free to add unit tests. \*\*If you're adding/amending code generation features then integration tests (\*end-to-end\* tests that use an example schema to generate and compile Java code) are essential\*\*. We can't merge pull requests unless they have integration tests, since without tests your feature could be lost in a future release.

## 4. Rebase and keep a clean log

Rebase against master often, if your pull request is stale it can't be merged. Rebase your commits into logical chunks, without errors (multiple commits are okay, but \*\*squash minor commits that fix errors in earlier commits\*\*). Force-push to your feature branch after you have rebased.