Contributing

-----------

There are two wonderful ways you can contribute to Canvas: filing issues and

submitting pull requests.

## Filing Issues

Filing a GitHub issue should be reserved for reporting bugs and defects in end-user functionality.

Feature requests and configuration issues do not belong here and will be closed.

See below for where to send those.

Likewise, this isn't a good place to make paid support requests, since the right people won't be alerted.

If you are one of Instructure's customers, please use the “Help” link in your Canvas instance.

If you do that you'll get a quick personalized response from our Support team.

When filing issues, we need to see details about what the problem is, what steps need to

be taken to reproduce the problem, and what you expect the behavior to be.

Since Canvas makes heavy use of different role types, one piece of data that should generally be included

in the reproduction steps is what role encountered the problem (e.g. student, teacher, admin).

It might be helpful to copy the following and use it as a template when writing up an issue:

```

Summary:

Steps to reproduce:

Expected behavior:

Additional notes:

```

We’ll try to get back to you in a timely manner and let you know if we need more details or not

and any status we can provide on expectation for a fix.

\*\*Feature Requests\*\* should instead be filed on our Canvas community site (https://community.canvaslms.com).

In order to log in and participate in the community you will need a Canvas account. If you don’t already have one,

the easiest way to get one is to go to http://www.canvaslms.com/try-canvas, and click “Build It” and register as a teacher for a free account.

\*\*Configuration Issues\*\* are generally best answered either on our user group mailing list or

in #canvas-lms on Freenode. See https://github.com/instructure/canvas-lms/wiki#getting-help

for a full list of options for getting help.

## Submitting Pull Requests

In order for us to continue to dual-license our Canvas product to best serve

all of our customers, we need you to sign our contributor agreement before we

can accept a pull request from you. After submitting a pull request, you'll see

a status check that indicates if a signature is required or not. If the CLAHub

check fails, click on Details and then complete the web form. Once finished,

the CLA check on the pull request will pass successfully. Please read our

[FAQ](https://github.com/instructure/canvas-lms/wiki/FAQ) for more information.

To save yourself a considerable headache, please consider doing development against

our master branch, instead of the default stable branch. Our stable branch is

occasionally reforked from master from time to time, so your Git history may get

very confused if you are attempting to contribute changesets against stable.

If you choose to contribute a pull request to Canvas, following these guidelines will make things easier

for you and for us:

- Your pull request should generally consist of a single commit. This helps keep the git history clean

by keeping each commit focused on a single purpose. If you have multiple commits that keep that focus

then that is acceptable, however "train of thought" commits should not be in the history.

- Your commit message should follow this general format:

```

Summary of the commit (Subject)

Further explanation of the commit, generally focusing on why you chose

the approach you did in making this change.

closes gh-123 (if this closes a GitHub Issue)

Test Plan:

- Use this space to enumerate steps that need to be taken to test this commit

- This is important for our in house QA personnel to be able to test it.

```

This format is the format that Instructure engineers follow. You could look at previous commits in the

repository for real world examples.

- The process your pull request goes through is as follows:

- An Instructure engineer will pull the request down and run it through our automated test suite.

They will report back with results of the testing. You can help this process along by running targeted

tests locally prior to submitting the pull request. You should also run `script/rlint` to make sure

your commit passes our linter.

- Once the test passes against our test suites, one or two engineers will look over the code and provide

a code review.

- Once the code review has been successful our QA engineers will run through the test plan that has

been provided and make sure that everything is good to go.

- If your commit touches any UI elements or behavior of the application, one of our product managers

will review the changes as well to make sure it is consistent with our product direction

- Once all these things have occurred then an engineer will merge your commit into the repository.

- Congratulations! You are now a Canvas contributor! Thank you for helping make Canvas great.

Guidelines

----------

- All new UI should be built in [React](https://github.com/instructure/canvas-lms/tree/stable/app/jsx) using the documented [API](https://canvas.instructure.com/doc/api/).

- Contributed code should pass our linters, but sweeping changes solely to correct lint errors in existing code should be avoided. The following scripts can be used to run the linters against changes in your code: `script/eslint` for JavaScript code, `script/rlint` for Ruby code, and `script/stylelint` for (S)CSS code.

## CLA

### What is it and why do I need to sign it?

The Instructure Contributor Agreement (ICA) is a contributor license agreement which grants us intellectual property rights in material you contribute to a project we own or manage. Signing the agreement creates a written record which allows us to track our legal rights when we use and distribute any project containing contributor material.

### What rights am I giving to Instructure?

When you sign the ICA you don’t give up ownership in the material. Instead, you are declaring that (a) you own the material you contribute, (b) that it doesn’t infringe on other intellectual property rights and (c) you are giving us joint-ownership or a license in the material.