## Open Development

All work on Blazorise happens directly on GitHub. All external contributors must send pull requests which will go through the review process.

## Setup Development Environment

To get the development and test environment set up on your local machine, you need to do the following:

1. Install Jekyll for documentation (https://jekyllrb.com/docs/installation/). If installing on Windows, follow these specific instructions: https://jekyllrb.com/docs/installation/windows/.

2. Install Node.JS (required for Selenium): https://nodejs.org/en/download/. Use the latest LTS, like node-v12.16.1.

3. Install Java SE (required for Selenium): https://www.java.com/en/download/. Use v8 or later.

4. Ensure recent build of Chrome browser installed: https://www.google.com/chrome/.

5. Install selenium-standalone: https://www.npmjs.com/package/selenium-standalone#install--run

You are now ready to build and test Blazorise:

### Running documentation

1. Start a command prompt and navigate to the docs folder under Blazorise root.

2. Run command: `bundle exec jekyll serve`.

3. Wait for build to finish and then open your browser and navigate to http://localhost:4000/

### Testing

1. Start a command prompt and run: selenium-standalone start. Note: this service needs to be running before you start Visual Studio, or test runs may fail.

2. Open the Blazorise solution at the root folder (Blazorise.sln).

3. Select Build > Build Solution on main menu. All of the projects should build sucessfully.

4. Run all of the unit tests in the solution use Test > Run All Tests on main menu. They should all pass at this point.

## Branch Organization

### Development

We use separate branches for development and for all new features. After all the features are done in the seperate branch it will be merged into the `master`.

Development branches follows the naming based on the milestone version number. For example if the milestone is `0.5.2` the main develop branch will be named `dev052` and all feature branches will be based on `dev052`. Feature branches follows the naming `dev052-feature-name`.

### Support

Similar to development branches we also have branches specifically for support. After new version is released the old `devNUM` branch will be removed and we will create new branch with the `sup` prefix and with the numbering from `dev` branch, eg. `dev052` > `sup052`

This branch is used ONLY for bug fixes. No new features can or will be added here.

## Pull requests

When submitting a pull request:

1. Fork the repository.

2. Create a branch from `dev{version-num}` and give it a meaningful name (e.g. `dev{version-num}-my-awesome-new-feature`) and describe the feature or fix.

3. Open a pull request on GitHub.