Developing

Overview

After you have completed all the necessary setup steps its time to start actually developing. The following items are necessary steps to getting your code added to the live site super group code as easily as possible.

For example purposes we are going to use "mytestgroup" as the group cname and "hubzero.org" as the hub we are working on. This would map to "hubzero" as the group name and "mytestgroup" as the project in Gitlab. We are also going to use "theuser" as the user's username in Gitlab.

Fork Project

The first step is you need to create a fork of the main project. You can find main project by navigating to your dashboard in Gitlab then the projects tab. Project names are formatted by the group/project, where group is the hub name/URL and project is super group cname.

1. Click on the project you want to start development for, you should be taken to the project page.
2. Click the "Fork repository" button on the right side of the page. This will fork the repository and take you to your forked version of this repository.

Clone Repository

You are now ready to clone the repository to a development machine. This can be anywhere, but recommended that you use the hubs dev machine or local dev machine (local HUB on VM).

1. Get the repository url. From your forked repository page you should see a text box with the git repo url in it. Copy that URL to your clipboard
2. Go to the machine where you want to clone the repository to and type the following into a terminal window:

   ```
   git clone git@gitlab.hubzero.org:theuser/mytestgroup.git; mv myte
   stgroup/* mytestgroup/.git* .; rmdir mytestgroup;
   ```

3. The repository content will be copied to a "mytestgroup" directory within the current directory

Add Upstream Repository

Upstream repository is a fancy word for the main repository you forked from. You need to tell your forked copy that it has a main repository and where it is. To add the upstream repository, in a Terminal window navigate to your cloned repo and type the following:

```
git remote add upstream git@gitlab.hubzero.org:hubzero/mytestgroup.git
```

You can test to see if everything was added correctly by typing:

```
git remote -v
```

and you should now see something like:

```
origin    git@gitlab.hubzero.org:testuser/mytestgroup.git (fetch)
origin    git@gitlab.hubzero.org:testuser/mytestgroup.git (push)
upstream  git@gitlab.hubzero.org:hubzero/mytestgroup.git (fetch)
upstream  git@gitlab.hubzero.org:hubzero/mytestgroup.git (push)
```

This is a very important part of working with Gitlab is keeping your forked repository synced with the main repository.

Develop

Make changes, add new code, fix bugs etc. Commit as you develop.

Sync with Main Project

Before you push your changes to Gitlab it is recommended that you sync your forked project with the main project.

Failure to sync your fork before pushing changes and creating a merge request can result in your merge request being denied until synced properly.

To sync, navigate to your cloned repo in a terminal window and type the following:

```
git fetch upstream
```

Then make sure your on the master branch by typing:

```
git checkout master
```

Then merge the upstream master branch with your master branch by:

```
git merge upstream/master
```

You might have to resolve some merge conflicts at this point. See the Git documation or search Google for issues you might run into.

Note: You can sync your forked project with the main project as often as you like. Syncing often usually reduces potential merge conflicts.

Push Changes

Pushing your changes it simple and easy. Simple type the following in a terminal window from within your cloned repo:

```
git push origin master
```

This pushes the changes you've committed to your forked projects repository.

Create Merge Request

A merge request is how the changes you pushed to your forked project get into the main project. Login to gitlab, go to your forked project, click the merge tab, then "New merge request". Select the master branch in your forked copy and click "Compare branches". You should be taken to the next step where you can give the merge request a title and a description. The description is very important for the approval team to understand what the merge is related to. This page will also show the commits that will be merged and the file diffs. When your ready click "Submit merge request".

Wait for Approval

Approval may be the next day or make take up to a week depending on complexity and schedules. Approvals are done Monday-Friday 8am - 5pm EST.

When your merge request is accepted or denied you will get an email notice regarding its status.

Pull Changes

Pulling in the changes that were merged into the main project can be done through the admin interface for the HUB. You must have admin rights to access the administrator interface.