

Git workflow documentation

Overview

Git is a decentralized version control system. GVSU University Libraries uses git for version control, to make it easy to collaborate on development projects, and to deploy code to our staging and production server (on Dreamhost).

Using git

Git is a command-line tool that must be installed on your local machine. You can install git at <http://git-scm.com/download> (available for Mac, PC, and Linux).

If you are using a Mac, be sure to download the Github Mac client, which makes pushing and pulling from our Github repositories easy. The client is available at <http://mac.github.com>

Creating a new project

Create a directory for your new project. Your local working directory for your project should mirror the eventual live version. Make sure your relative paths match local and live (especially if you are using mysqlconnect.php).

Now make it a git repository:

```
> cd /project_name
> git init
```

This creates a 'master' branch for your project. Master should always be ready to go live—you don't want to have code that doesn't work in this branch. You don't want to edit the master, that's like editing files live. You want to make a development branch that you will work in.

```
> git branch develop
```

You can see what branches you have created and the current working branch with the command:

```
> git branch
* master
develop
```

In this case, master is the currently selected branch. You need to checkout the develop branch to start working:

```
> git checkout develop
> git branch
master
* develop
```

Now you can start creating files in your project. When you are ready to commit them to your git repository, add them to the queue:

```
> git add .
```

If you only want to commit certain files, replace the '.' with the filename(s). If you want certain files to be ignored (especially files with passwords, etc.) then you might want to create a .gitignore file. You can do this manually or within the Github for Mac client.

When you are ready to commit, check to make sure all the changes are queued:

```
> git status
```

And then commit, adding a commit message that describes the changes you've made.

```
> git commit -m "First commit"
```

If you need a more comprehensive view of the changes that have been made, especially across branches, use

```
> gitk
```

Pushing your project to Github

We have an organizational account on Github, which allows us to have many different users collaborate on our coding projects. Because all of our code in Github is public, we want to be especially sure that we never include any files in our git repositories that contain passwords or other sensitive information.

You'll need a Github account of your own, which you can set up at <http://github.com> (just do the free account). You will then be added to the GVSULib organization. You'll need to set up your ssh public keys, which can be tricky. Follow the directions on the Github site.

The easiest way to get a new repository on Github is with the Github for Mac client. Simply add the local repository to Github for Mac and then Publish it to our gvsulib Github. You're done! Now your coworkers can clone this to their local machines and collaborate with you.

To use the command line, create a new repository in Github and then follow the directions they give you to add Github as a remote and push using the shell.

Getting a project off Github onto your machine

To start collaborating with your coworkers on a project, you need to clone a repository from our gvsulib Github account. The easiest way to do this is in Github for Mac. In the left sidebar, select "gvsulib" and then find the repository you want to work on. Click "Clone to computer" and put the project in your root development directory. That's it! Make sure to check out the develop branch before you start working. You can push your changes to the develop branch on Github.

To use the shell, just clone the repository on your machine.

```
> cd /root_development_directory
git clone git@github.com:gvsulib/DIRECTORY-NAME.git
```

That's it. You can find the path to the directory name on the Github repository page, right above the list of files.

Deploying code with git

Once a project is on Github, you can deploy it to our Dreamhost server by sshing into Dreamhost and using git. For instance, to launch a site at gvsulib.com/labs/awesomesauce, you'd clone the Github repository on Dreamhost like this:

```
> ssh *****@*****
cd /labs
git clone git@github.com:gvsulib/DIRECTORY-NAME.git
```

That's it. If your relative paths are all set, the site should function exactly as it does on your local machine. If you make more changes locally and push them to Github (on the master branch), you can push these changes to the live site like this:

```
> ssh *****@*****
cd /labs/DIRECTORY_NAME
git pull origin master
```

And you're done.