Website Conversion

lab #lab03

James L. Parry
B.C. Institute of Technology

Lab Goals

The purpose of this lab is to introduce you to CodeIgniter and the workflow for working with webapps.

This lab is *not* indicative of the process you want to use when in "full development" mode ... it has you share a single repository, with both members of a team having write access - a recipe for disaster! I set the lab up this way to break you in gently to good collaboration practices - if you mess things up now, it is relatively easy to recover :)

Suggestion: you may want to skim the slideshow first, before working your way through it. I have included some hints for dealing with repository merge conflicts, but they are only a start.

Lab Submission

Your lab will be completed in a github repository.

Submit a readme to the lab dropbox, with a link to your github repository.

Due: three days after your lab, eg. Thursday lab period will have theirs due Sunday, Jan 24, 17:30 PST

This is a pair lab, sharing a single repository, but each using their own github account and computer.

If you insist on working on your own, you will need to setup a second github account, and use two machines, to get the full benefit (and marks) of the lab.

Lab Preparation

You prepared for this lab last week, with:

- your D2L group
- your github setup
- your cloned repo
- your AMP setup
- your CodeIgniter starter
- all together in your repo

This lab will take you through the conversion of the provided static website into a CodeIgniter webapp :)

Prepared?

You should have your `htdocs` folder ready to roll, with the CodeIgniter `system3` framework folder inside it, as well as your lab's project folder.

You will have a local domain defined and virtual hosting setup in your Apache.

You may have the CodeIgniter user guide in a folder on your desktop for convenience.

Lab Work

There are three tutorials to complete for this lab, for marks... Don't start yet!

- Tutorial ci-basic01: trivial CI site
- Tutorial ci-basic02: basic CodeIgniter webapp
- Tutorial ci-basic03: good CodeIgniter webapp

Lab Process - the fun part...

- You have a shared repo from last week. One of you owns it and the other is a collaborator. You both have push rights to this repo. Each of you will clone your shared repo to their respective machines.
- Complete the first tutorial with one of you "driving" and the other "navigating". When done, make sure you (the driver) commit and push your changes to your shared repo.
- Before starting the second tutorial, switch roles and computers. The new "driver" will need to synchronize their repo with the shared one, and then work through the tutorial. Remember to commit and push when done.
- Before starting the third tutorial, switch roles and machines again. Synchronize, complete the tutorial, and push/commit your changes.
- Both of you should synchronize your local repositories with the shared ones, once the tutorials are completed.

Synchronizing Backstory

Synchronizing repositories is the process of making sure they start from a consistent codebase.

If the shared repository has changed since you last synchronized, you run the risk of your local changes being rejected when you try to push them to the shared repository.

Some of you will find it easier to work with github from within your IDE, while others might prefer a command line (git bash shell). Some commands are easily done from the command line.

To make your life a bit easier, I have posted a couple of cheat sheets to the "supplementary" module in our D2L content. The "github" one is from the horse's mouth, while the "git-tower" one has some helpful hints & explanations.

Synchronizing Procedure

The following *local* steps are an appropriate procedure to synchronize repos:

- Before you start a new coding session, synchronize repos by doing a `git pull origin`, either from a git bash shell (in your project folder) or by using the appropriate "git pull" icon in your IDE toolbar. This will fetch any changes from the shared repo, and merge them into your local project. If there are merge conflicts, resolve them before continuing.
- Code away. Commit related changes, with good commit comments.
- When you have completed a logical chunk of code, and it is working, you are ready to push your changes to the shared repository...
- Commit any outstanding changes. Resynchronize with the shared repo, resolving any merge conflicts. Push your committed changes to the shared repo, by doing a `git push origin`, from either the command line or inside your IDE.

Resolving Merge Conflicts

In spite of your best intentions, collaborators get their repositories out of sequence with the shared repository. When this happens, you have two choices: fix the inconsistencies, or start over (disaster recovery).

Git has a mergetool to help resolve conflicts. It presents inconsistencies between two repositories, and lets you choose which version to keep. Any changes managed this way need to be commited locally before your repository is deemed conflict free.

Disaster Recovery

If the repository inconsistencies are too extreme for the mergetool to fix, then you start over. You could copy or rename your local repository, and then reclone the shared repository and re-apply the changes you intended to make. There is a `git stash` tool to help with this.

Lab Marking Guideline

This lab will be marked out of 10, as follows:

- Tutorial basic01: trivial CI site (3)
- Tutorial basic02: basic CodeIgniter webapp (3)
- Tutorial basic02: good CodeIgniter webapp (2)
- Process - proper commit history (2)

Your github repository will be in a state consistent with the end of one of the above, and you do not need three repos!

Disclaimer

We will not necessarily be following "best practices" throughout this lab.

Our focus is on "baby steps", that will lead to best practices over the next few labs.

Congratulations!

You have completed lab #lab03: Website Conversion

If you would take a minute to provide some feedback, we would appreciate it!

The next activity in sequence is: ci-basic01 Static to Trivial CI

You can use your browser's back button to return to the page you were on before starting this activity, or you can jump directly to the course homepage, organizer, or reference page.