Models

lab #lab04

James L. Parry
B.C. Institute of Technology

Lab Goals

The purpose of this lab is to help you explore and practice some of the model techniques from lesson 4.

There are two parts to this lab: A) the basics and B) an installer.

You only need complete part A to get a good mark. Part B is specifically for those looking for more of a challenge, and I suggest you only tackle it after successfully completing part A.

Suggestion: you may want to skim the slideshow first, before working your way through it.

Lab Submission

Your lab will be completed in a github repository. This will be your (forked) winter2015-lab04 repo. If you are doing the optional part of the lab, your repo will include the Part B changes, and you should indicate that you are shooting for the optional marks in your readme.

Please make sure that your work is in the "master" branch or your repository, and that any other branches have been merged into it.

Dayschool students: submit a readme to the lab dropbox. This readme should have a link to your github repository.

Distance students: send me an email with a link to your github repository.

Due: Weds, Feb 11, 17:30 PST

Lab Marking Guideline

Part A addresses three models, and the corresponding changes to our two controllers.

The menu and orderitems models are trivial, and effectively done. The orders model is where most of the action happens. The orders model has five methods needed, for 1 mark each.

Making the welcome and order controllers work properly will get you 1 and 2 marks, respectively.

That adds up to 8 marks. Part B is worth 2-3 marks, depending on the elegance of your solution, and how easy it would be to use as an example. Simply taking a stab at part B, without solving the challenge, won't get you any marks!

Commenting: add comments sufficient to convince the reader that you know what you are doing. Inappropriate/inadequate comments will cost you up to two marks off.

Part A: The Basics

Complete tutorial 4.

Part B: Installer

Build an installer "system" to include in a webapp.

The purpose of the installer is to configure a freshly "installed" application properly for the local database settings.

It should prompt for a username and password that have appropriate permissions to use with MySQL, and for the name of the database to use for local deployment.

It will then need to modify the application/config/database.php to reflect the settings, and then, using the database forge & utilities for CodeIgniter, setup the database properly. This should create and populate the tables needed for the webapp.

Part B Notes

Part B is a challenge, on purpose. Don't even try this until you have completed part A, and then only if you are willing to dive a bit deeper into CodeIgniter and hooks.

The app should start out with innocuous config files autoload.php and database.php. Once you have the above data, and have setup the database, then tailor the two "real" config files appropriately, so that the "real" webapp runs properly once the installer is exited. An easy way to do this would be with some "views", with substitution fields for the prompted for data.

How do you know if the app has been installed? You will need a marker file of some sort, perhaps the "views" just alluded to. If present, the webapp has not been "installed"; if absent, there is nothing for you to do.

Congratulations!

You have completed lab #lab04: Models

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

The next activity in sequence is: lesson06 Helpers and Addons

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.