Week 3 Section

Demo MVC, RESTful Routes and CRUD w/ Sinatra

Quick setup:

```
git clone https://github.com/jeremywrnr/sinatra-intro
cd sinatra-intro
bundle install
ruby template.rb # OR: bundle exec ruby template.rb
```

Then open this webpage:

```
http://localhost:4567/todos
```

Also try with `curl`:

```
curl http://localhost:4567/todos
```

Instructions

This section we will take a look at how to apply ideas of MVC, RESTful Routes, and CRUD in the context of the Sinatra framework to build a to-do list app. When you're done, users should be able to go to your website, view their list of to-do items, create new list items, edit list items, and delete list items. We will be building the codebase together so pair up and get the starter code at:

https://github.com/jeremywrnr/sinatra-intro

Task 1

The first thing we are going to do is create a model. Unlike Rails, Sinatra doesn't have MVC baked in so we're going to hack our own. We're going to use ActiveRecord on top of a SQLite database. In this application, what is our model going to be, and what CRUD operations are we going to apply to the model?

- (a) index:
- (b) create:
- (c) read:
- (d) update:
- (e) destroy:

Task 2

Next, let's create some routes so that users can interface with our app. Here is an example URL: `https://www.etsy.com:443/search?q=test#copy`

Break down the URL into its component parts:

- https:// :
- etsy :
- 443 :
- /search :
- q=test :
- copy :

In Sinatra the routing and controller are coupled. It's easy to declare paths. We're going to use declare some RESTful routes so that we can view a list of to-do items, create a to-do item, edit a to-do item, and delete a to-do item. What RESTful actions should we use for these?

Task 3

Since HTTP is a RESTful protocol, every request must follow with a response, so we need to return a view or redirect to every request. We're going to use JSON for our responses, which is similar to what a lot of APIs do. Where should the response go?

Reference: http://sinatrarb.com/intro.html