Module 4
MVC, RESTful Routes, CRUD, Sinatra

CS W169A: Software Engineering

John Yang | Summer 2020

1 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'redone, 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, with the starter code located at https://github.comjohn-b-yang/cs169-exercises. If you are able, find a classmate to pair program through these exercises! Here is the reference to Sinatra, which will be helpful!

2 Setup

Open a command line prompt and enter the following commands:

```
git clone https://github.com/john-b-yang/cs169-exercises
cd cs169-exercises/sinatra-intro/
bundle install
ruby template.rb # OR: bundle exec ruby template.rb
```

Then, enter the following link into a browser to view the webpage and check if it's working.

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

Also, try this following command using 'curl' to verify that the app is running locally and responsive. The command fires a GET request to retrieve the "to do" list, and you should receive a response displayed in the command line's standard output.

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

In the following exercises, we'll be adding more routes, and you can continue to use curl commands with different arguments to verify their behaviors' correctness.

Goal: Your task is to implement the parts of the file labeled "YOUR CODE HERE". The reference containing the solutions is in the final.rb file.

3 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?

- index:
- create:

- read:
- update:
- destroy:

4 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/search?q=test#copy`

First, specify what parts of the URL are which components based on our discussion of the anatomy of a URL from lecture. (If this doesn't ring a bell, review Module 4.4: Routes, Controllers, Views). Check out this post by IBM detailing the components of a URL.

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

In Sinatra the routing and controller are coupled, making it easier 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?

5 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?