Lab Assignment #6

Fall Semester 2019

April 17rd

Repository available at: https://classroom.github.com/a/9eqp4MB2

1 Introduction

This sixth lab assignment will be graded and you will be working by yourself. We expect from you to not collaborate with others during the lab. Your work must conform to the *Code of Academic Ethics* included at the end of this document.

You must hand in your work by Saturday, April 20th at 23:59 hrs. in Git.

2 Description

You will have to develop the registration and login service on an existing ruby on rails platform, using a base project that you will find inside your repository, once you accept the GitHub classroom invitation.

So, for being able to complete this assignment you will have to apply your knowledge about Models (ActiveRecord), Controllers (ActionController), Routes and some basic rails functionality . We will provide you with some predefined views in order to use the RoR platform through the browser. This means that you will not have to edit any of the views that we are providing to you, instead, you will have to edit models, controllers and routes.

3 Auth, Login and Registration Flow

In a basic web application, the interaction flow between the client and the server usually involves the client trying to access resources or data that need to be protected/forbidden from unwanted clients/users. This is achieved by having the server grant or deny access to the client, after a login action. When the user logs in, the server will send a piece of information (user id for simplicity) attached in the HTTP header of the response, telling the web browser to store that piece of information and send it back to the server in every successive request to the web application. This is basically what you learned in the previous class, i.e., the header field `SET-COOKIE` is used to accomplish this. Next, the web browser will send your cookie alongside https requests back to the server, where the cookie (and user id) can be verified to take corresponding actions.

For a login flow to work, we will need to send an http request to the server from the client, with some credentials (e.g., username, password) and wait for the server to give us the green light, or the red one. This means, that we will not be able to access some routes/paths of our application if we are not correctly logged in. Once we are authenticated and accepted by the application, we will be able to access protected content. There are different ways to implement an authentication flow or layer, but for the matter of this assignment we will start with a basic approach, using session cookies implemented by the Rails framework through the session helper.

The registration flow works similarly, but in this case the application will first create a record for the user information, received through a registration form, and then redirect the user to the login view to start the login flow.

4 Goals

Inside the base project you will find four controllers that you will have to modify and complete with code. First, the `UsersController`, in charge of dealing with user-related actions (all actions must be authorized before being called). Second, the `SessionsController`, in charge of dealing with authentication of user credentials through the login view. Third, the `RegistrationsController`, which has to be able to register a new user by obtaining data entered through a view. Finally, the `ApplicationController` is the main controller in a Ruby On Rails application, and can be used to trigger some callbacks or filters before any actions occurs.

The next list of items is what you have to accomplish by the end of the lab assignment:

1. Complete the registration for an user [1.5 pts]
   - Registration controller: complete the `create` method to create new user.
     - check for all the fields required (name, lastname, email, password, phone).
     - redirect to the login view in case of successful registration, therein displaying a message indicating success (which must be passed in by means of a flash variable), or,
     - redirect to the registration view in case the user registration validations fail. The registration view must display an error message provided through a flash variable.
2. Complete the login flow for a platform user [3 pts]
   (a) Application controller: complete a `current_user` helper
   (b) Application controller: complete a `is_user_logged_in?` helper method
   (c) Session controller: complete the `create` method to set logged user
      i. redirect to user show view in case of approved log in, with flash success message.

ii. redirect to login view in case of rejected log in with flash error message.

For this method take a look at the useful links for the Rails Session guide. For storing sessions persistently we will be using session cookies.

(d) `SessionsController`: complete the `delete` method for user log out

3. Complete `UsersController`: [1.5 pt.]

   (a) complete the `create` method. It must set a flash message for success and error.

   (b) complete the `update` method. It must set a flash message for success and error.

5 Grading

Your grade will be computed based on your accomplishments, considering the scores listed above for every milestone. Consider six points plus the base one.

6 Testing your application

To test your application you can run the rails server, and issue requests from a browser by entering URLs that map to your defined routes and actions.

7 Useful links

The following links to Rails Guides will provide you useful information for completing your assignment:

- Command line: `http://guides.rubyonrails.org/command_line.html`
- Rails Routing: `guides.rubyonrails.org/routing.html`
- Rails Action Controllers: `http://guides.rubyonrails.org/action_controller_overview.html`
- Rails Rendering (onlyfor JSON render): `http://guides.rubyonrails.org/layouts_and_rendering.html`
- Rails Session: `http://guides.rubyonrails.org/action_controller_overview.html#session`
- Rails Filters: `http://guides.rubyonrails.org/action_controller_overview.html#filters`

8 Repository Setup

First, you need to sign up at the classroom assignment on Github, following the same steps as last week. Go to the Github classroom assignment at https://classroom.github.com/a/9eqp4MB2(copy and paste this URL in your web browser) and accept the invitation to the assignment.

After your personal repository has been created, do a git clone specifying the URL provided by Github, to your Desktop location in you computer. You will find the clone URL in the button "Clone or Download" located at your repository profile.

Please, remember to properly set up your working environment for Git by running the following commands in your terminal window (of course, fill in the parameters with their actual values):

```
$ git config --global user.name "your name"
$ git config --global user.email "your email address"
```

9 Code of Academy Ethics

During the course and for every task, test and homework given to you, all the ethical criteria established by the Faculty of Engineering and Applied Sciences at the Universidad de los Andes, Chile will apply:

> "Any detection of copying, plagiarism, or dishonest behavior, independent from the fact itself, will be reviewed by the Faculty Council. The minimum sanction will be a 1.0 as the final grade of the course, with the possibility of escalating to the expelling of the student from the University.
>
> Any student suspicious of unethical behavior will be punished as if he/she had committed or executed the suspected dishonest behavior. In other words, it is exclusively the responsibility of the students to behave correctly and ethically."