Cloud Application Development

The V and C in MVC. Routing. Problem

In this lab we will be creating a simple Rails application to perform arithmetic operations. Application functionality:

* The application asks the user to provide two numbers, and to select the arithmetic operation to be performed. The application will then display the result of the arithmetic operation.
* What are the main tasks of the application? How can we implement them?

1. Create a new rails application called ***calculatorApp***. At the terminal type

# rails new calculatorApp

1. Recall that we must navigate (by using the command cd to the root folder of our application (i.e. calculatorApp in this case)

* cd calculatorApp
* rake db:migrate

1. In another terminal start the web server to run the application. Recall that before we start the web server, we must first navigate to the root/ parent folder of the Rails application. At the terminal type the following commands:

* Navigate to the root of your application
  + cd calculatorApp
* Start a web server to run the application
  + rails server --binding 0.0.0.0
  + If you run Rails locally on your own machine, then you start the web server by typing at the terminal rails server

1. Let’s implement the application’s functionality
2. First, let’s write the ruby code that performs the arithmetic operations
   * Create a ruby file in the lib directory of the Rails application named

arithmetic\_calculator.rb

* + In the arithmetic\_calculator.rb declare one method for each arithmetic operation. A possible solution is presented below.

•

class ArithmeticCalculator def self.add(n1, n2) result = n1 + n2

end

def self.subtract(n1, n2) result = n1 - n2

end

def self.multiply(n1, n2) result = n1 \* n2

end

def self.divide(n1, n2) result = n1 / n2

end end

1. How would a user have access to this application functionality? We should choose a URL through which the functionality to perform arithmetic operations is available to the users. Let’s suppose that the suffix to be added to our application’s root URL is /calculate
2. In browser append immediately after the http://<ip addresss>:3000/calculate
   * What is the result?
     + You should see an error message *“No route matches [GET] "/calculate"”*
   * Why do we get this error?
     + **Answer:** There is no route in the *routes.rb* file which matches the pattern of the above URL. Therefore, Rails does not know which controller and which corresponding action has to serve our request.
   * We can inspect all the routes defined in our application using the command rake routes. Type the aforementioned command at the terminal. What do you observe?
   * Does the application have any controller? No. To generate an empty controller, named *calculator* without any action, type at the terminal
     + rails generate controller calculator
   * Add one of the next 2 lines (note that they are equivalent) to the *routes.rb*

file (located in the *calculatorApp ! config* folder)

* + - get '/calculate', :controller=>'calculator', :action=>'calculate'

or

* + - get '/calculate' =>'calculator#calculate'
    - The above information tells Rails that the controller called *calculator* (located in *app !controllers !calculator\_controller.rb*) is going to server our request by using the action called *calculate*
  + Refresh the browser page (i.e. http://<ip addresss>:3000/calculate)
  + What is the result?
  + You should see an error message *“The action ’calculate’ could not be found for CalculatorController”*
  + Why do we get this exception? Does the action exist? Let’s implement the action in the *calculator\_controller.rb*:

class CalculatorController < ApplicationController def calculate

end end

* + Refresh the browser page (i.e. http://<ip addresss>:3000/calculate)
  + What is the result?
    - You should see an error message “Missing template calculator/calculate..."”
  + Why do we get this error?
    - Answer: There is no view defined for the calculate action. By default, the view with the same name as the action is rendered.
  + Next, we create the Input View template: a page that will ask the user to input two numbers. Q: Where should this page be created?? A: In the app ! views folder
    - Create a file in the app ! views ! calculator folder of the Rails application. Name the file calculate.html.erb
    - What elements should our page display??? We should provide 2 text fields for the user to input the numbers

\* Add the following content to the above file

<h1>Simple Calculator</h1>

<%= form\_with url: "/sum", local: true, data: { turbo: false } do |f| %>

<%= text\_field\_tag :number1 %>

<%= text\_field\_tag :number2 %>

<%= submit\_tag "+" %>

<% end %>

* + - Note, above the /sum specifies the URL (i.e. the action that should happen when the form is submitted) where to send the form-data when a form is submitted
  + Refresh the browser page (i.e. http://<ip addresss>:3000/calculate)

1. Now, the page that we have just created should be visible. Let’s try to calculate the sum of two numbers. Enter two numbers and press the **+** button. What do you observe?
   * There is another error “No route matches [POST] "/sum"”
   * Why do we get this error?
   * Answer: Rails does not know which controller and action has to serve our request because there is no route in the routes.rb file to match the generated URL (i.e. /sum).
   * Remember that in the calculate.html.erb we have a line

The

<%= form\_tag "/sum" do %>

form\_tag

uses /sum to specify where to submit the form data. We have to connect this to an action/method from the controller that does the addition of the two numbers, and therefore we have to modify the routes.rb file.

* + Add one of the next 2 lines (note that they are equivalent) to the routes.rb file (located in the calculatorApp ! config folder)
    - post '/sum', :controller=>'calculator', :action=>'sum'
    - or
    - post '/sum' =>'calculator#sum'
    - The above information tells Rails that the controller called calculator is going to server our request by using the action called sum
  + Refresh the browser page (i.e. http://<ip addresss>:3000/sum)
    - Furthermore, the controller does not contain an action which is able to calculate the sum! This is what we have to implement next.
  + Let’s modify the calculator\_controller.rb by adding the method sum. The method sum is the action that the calculator controller has to take when adding two numbers. We are going to use the logic we defined in the arithmetic\_calculator.rb.

def sum

# the user data is sent via the params

# :number1 key contains the first number entered # :number2 key contains the second number entered # the numbers are entered as a text

# retrieve the number submitted by the user @num1 = params[:number1]

@num2 = params[:number2]

# we use the arithmetic\_calculator.rb of the lib folder @result = ArithmeticCalculator.add(@num1.to\_i, @num2.to\_i)

end

* Let’s tell the controller where to find the ArithmeticCalculator’s definition. At the beginning of the *calculator\_controller.rb* file add the next line

require 'arithmetic\_calculator'

1. Let’s try again to calculate the sum of two numbers and press the + button. What do you observe?
   * There is another error “Missing template calculator/sum ”
   * Why do we get this error?
     + Answer: By default, a view (i.e. a .html.erb file) with the same name as the controller’s action/method is rendered at the end of the execution of the controller’s action’s body. In this case the expected view is sum.html.erb
2. Create the Output View template to inform the user the result of the operation. Q: Where do we have to create this file??? A: Create the file in the app ! views ! calculator folder of the Rails application. Name the file sum.html.erb
3. What should be the information displayed on the newly created view?? Add the following content to the sum.html.erb file

<h1>Sum</h1><br>

<%= @num1 %> + <%= @num2 %> = <%= @result %>

1. Run the application again, by loading the /calculate suffix (i.e. http://<ip addresss>:3000/calculate)
   * Now we can test the functionality of our application: perform some additions.
2. Let’s do one more change: update the content of the sum.html.erb file to look as below, namely, we include in the view also the content of the input view (i.e. calculate.html.erb)

<h1>Simple Calculator</h1>

<%= form\_with url: "/sum", local: true, data: { turbo: false } do |f| %>

<%= text\_field\_tag :number1 %>

<%= text\_field\_tag :number2 %>

<%= submit\_tag "+" %>

<% end %>

<hr>

<h1>Sum</h1><br>

<%= @num1 %> + <%= @num2 %> = <%= @result %>

1. Run the application again

# Task Extend the application to allow subtraction, multiplication and division. Resources

* + Form Helpers
    - <http://api.rubyonrails.org/classes/ActionView/Helpers/FormHelper.html>
  + Form Tag Helpers
    - [http://edgeapi.rubyonrails.org/classes/ActionView/Helpers/FormTagHel](http://edgeapi.rubyonrails.org/classes/ActionView/Helpers/FormTagHelper.h) [per.h](http://edgeapi.rubyonrails.org/classes/ActionView/Helpers/FormTagHelper.h)
  + Layouts and Rendering
    - <http://guides.rubyonrails.org/layouts_and_rendering.html>