

Exercise 1 Setting up the Virtual Machine

SDM 409232 2015

This exercise guides the use of Bootstrap to make some low effort programmatic changes that result in a better user interface by making use of the component library in Bootstrap <http://getbootstrap.com> .

It is also worth using the icons and other scalable components in the Awesome Fonts gem <http://fortawesome.github.io/Font-Awesome/> (left as an exercise for the student) .

The use of “.haml” files to inject Ruby code into HTML using templates is also introduced <http://haml.info/tutorial.html> . These replace the “.erb files” in the rails project (embedded ruby). Note that the embedded Ruby in either file can return html in-place to the client browser, or just invoke code run on the server-side only.

To see the differences between the user interface with and without the use of Bootstrap, compare:

BEFORE: <http://wpm4240.herokuapp.com/movies>

With AFTER <http://wpm4240new.herokuapp.com/movies>

How to create the second version of Rottenpotatoes

Rotten Potatoes with Bootstrap

1. Installation of Bootstrap Gem

Open GemFile

Add new line below the rails gem

```
gem 'bootstrap-sass', '2.1'
```

Save file and close

2. Update Stylesheet Tree

Navigate to app>assets>stylesheets

Open **application.css**

Add the following lines into the comment section at top of the file

```
*= require_self
```

```
*= require_tree
```

Comment out or delete all current styling in the file

3. Create SCSS file in the stylesheets folder

```
custom.css.scss
```

Open the file

Add import statement for bootstrap

```
@import "bootstrap";
```

4. Run bundle install from app root folder to update with the new gem

```
bundle install
```

Text Editor

You may use any text editor in Ubuntu to make these changes. The default editor with your virtual machine is “gedit” and this is fine. Sublime-Text is another editor commonly used by developers and version 3 is now available for free. Note that changing to the application directory and typing “subl .” will open sublime text editor and show all the application directories/files for quick editing.

Type the following changes rather than copy and paste, to get an idea of how sensitive Rails is!

Add the following Snippet of code for **app>assets>stylesheets>application.css**

```
1 ▾ /* Simple CSS styling for RottenPotatoes app
2  *= require_self
3  *= require_tree
4  */
5
6
7  /*rest the top bar to the top of browser*/
8 ▾ body{
9     margin-top:0;
10 }
11
12
13 ▾ .header{
14     margin-top:0;
15     background-color: #4f81bd;
16     padding-left:10px;
17     padding-top:10px;
18     padding-bottom:10px;
19     text-align:left;
20     color:#FFFFFF;
21 }
22
23 /*set box margin of the whole page*/
24 ▾ #main{
25     margin-left:10px;
26     margin-right:10px;
27 }
28
```

Add the snippet of code for new file **app>assets>stylesheets>custom.css.scss**

```
1 |@import "bootstrap";
2
```

HAML Indentations

Indentations are 2 more space than its parent element (TAB not to be used). YOU WILL SEE ERRORS IF YOU GET THIS INDENTING INCORRECT.

Note that start and end tags from HTML are replaced by % tags and indenting, in haml files – faster when you get used to it?

Change the **app>views>layouts>application.html.haml** file with the following snippet of code

```
1  !!! 5
2  %html
3    %head
4      %title Rotten Potatoes!
5      = stylesheet_link_tag 'application'
6      = javascript_include_tag 'application'
7      = csrf_meta_tags
8
9    %body.mainContainer
10      .headerContainer
11        %h1.header Rotten Potatoes!
12      #main
13        - if flash[:notice]
14          #notice.message= flash[:notice]
15        - elsif flash[:warning]
16          #warning.message= flash[:warning]
17
18      = yield
```

Snippet of code from `app>views>movies>edit.html.haml`

```
1 |# in app/views/movies/edit.html.haml
2 %ol.breadcrumb
3   %li
4     = link_to 'All Movies ', movies_path
5   %li.divider /
6   %li
7     = link_to 'Movie Details', movie_path(@movie)
8   %li.divider /
9   %li.active Editing
10
11 = form_tag movie_path(@movie), :method => :put do
12
13   = text_field :movie, 'title'
14
15   = label :movie, :rating, 'Rating'
16   = select :movie, :rating, ['G','PG','PG-13','R','NC-17']
17
18   = label :movie, :release_date, 'Released On'
19   = date_select(:movie, :release_date, order: [:day, :month, :year])
20
21   .form-group
22     = text_area :movie, :description,
23       :rows => 5,
24       :class => "form-control",
25       :style => 'width:350px'
26
27   .form-group
28     = submit_tag 'Save Changes',
29       :class => "btn btn-success btn-large"
30
```

Breadcrumb linking

.breadcrumb is a class that is defined in bootstrap

.active to change colour of text to grey showing what page we are currently on

order: specifies the order of occurrence of the date objects DD-MM-YYY on the page

text_area new element to get a description of the movie

btn btn-success btn-large class defined in bootstrap which creates a button which is green

Snippet of code from `app>views>movies>index.html.haml`

```
1 |# This file is app/view/movies/index.html.haml
2 %ol.breadcrumb
3   %li All Movies
4
5 %table#movies.table.table-hover
6   %thead
7     %tr
8       %th Movie Title
9       %th Rating
10      %th Release Date
11      %th More Info
12   %tbody
13     - @movies.each do |movie|
14       %tr
15         %td= movie.title
16         %td= movie.rating
17         %td= movie.release_date.strftime("%d-%b-%Y")
18         %td= link_to "More about #{movie.title}", movie_path(movie)
19
20 =link_to 'Add new movie', new_movie_path, :class=>"btn btn-success btn-large"
21
```

strftime haml function which lets you control the format of date output

Snippet of code from `app>views>movies>new.html.haml`

```
1 |# in app/views/movies/new.html.haml
2 %ol.breadcrumb
3   %li
4     = link_to 'All Movies ', movies_path
5   %li.divider /
6   %li.active New Movie
7
8 = form_tag movies_path, :method => :post do
9
10  .form-group
11    = text_field :movie, :title, :class => "form-control ", :placeholder => "Title"
12
13  .form-group
14    = label :movie, :rating, 'Rating'
15    = select :movie, :rating, ['G','PG','PG-13','R','NC-17']
16
17  .form-group
18    = label :movie, :release_date, 'Released On'
19    .form-control
20      = date_select(:movie, :release_date, order: [:day, :month, :year])
21
22
23
24  .form-group
25    = text_area :movie, :description,
26      :rows => 5,
27      :class => "form-control",
28      :style => 'width:350px',
29      :placeholder => "Give A Description"
30
31  .form-group
32    = submit_tag 'Save Changes', :class => "btn btn-primary btn-large"
33
```

USE GIT and Heroku

Remember to commit your changes to your local git repository and push the changes to your on-line repository in github. Then deploy your application to the cloud using Heroku and check it works in a browser anywhere.

For working in teams you will generally invite collaborators to a single repository and create branches for work you are doing (the master should be deployable at all times). Have a look at some recommendations for team workflows with git (eg <https://guides.github.com/introduction/flow/index.html>)

You should be able to create a new rails application with a database with models, views, an controllers. Use github, and deploy it to heroku. You should understand the purpose of a Gemfile, css files (and scss), Bootstrap, haml files. You should understand the purpose of “migrate”, “bundle” and “scaffold” in Rails.

You should understand how to use these **Linux commands** (at least): `cd`, `cd ~`, `cd ..`, `ls`, `rm`, `mkdir`, `gedit`

You should understand these **git commands**:

`git init`, `git status`, `git diff`

`git add`, `git commit -m “message”`,

`git branch branch_name`, `git checkout branch_name`, `git merge branch_name`

`git pull`, `git push`

Example making new application: OPTIONAL EXERCISE IF NEED MORE PRACTICE

CREATE AN APPLICATION THAT STORES THE NAMES EMAILS, AND DESCRIPTIONS ABOUT MEETINGS WITH EXPERTS.

Log on to your virtual machine development environment and start typing:

mkdir new_app

cd new_app

rails new new_app */* look at all those directories created – you should understand the purpose of most of these */*

rails server */*starts WEBrick server. Look at the welcome page from Rails using a browser “localhost:3000”. <ctrl-c> to stop the server.*

git init

git add .

git commit -m “initial commit for new_app”

git remote add origin git@github.com:yourgithubusername/new_app.git

git push -u origin master

git checkout -b experts */* create and switch to a new branch called “experts */*

NOW THE MAGIC....

rails generate scaffold Expert name:string email:string

bundle exec rake db:migrate

git commit -m “Experts application created”

rails s */* a short cut for “rails server” */*

On a browser explore the application:

localhost:3000/experts

localhost:3000/experts/new

create some entries for experts (enter a few names and emails).

On a browser, explore the app. Have a look at

localhost:3000/1

localhost:3000/1/edit

localhost:3000/new

Stop the server and have a look at the model, view and controller files created using an editor such as gedit.

Model: /app/models/expert.rb

Controller: /app/controllers/experts_controller.rb

View: /app/views/experts/

Now add another table in the database to keep short descriptions of last contact with each expert

rails generate scaffold Meeting description:string expert_id:integer

bundle exec rake db:migrate

Start the rails server and have a look at the changes to the application on a browser

commit the changes to git (locally)

Now make an association between each expert and multiple meetings.

Change the app/models/expert.rb file:

class Expert < ActiveRecord::Base

attr_accessible :email, :name

has_many :meetings

end

change the app/models/meeting.rb

class Meeting < ActiveRecord::Base

attr_accessible "description, :expert_id"

belongs_to :expert

end

Start the rails server and have a look at the changes to the application on a browser

Commit your code locally using git

Merge the "experts" branch back in to the master in the "new_app" repository

git checkout master /*switch to the branch "master" */

git merge expert /* merge the "expert" branch with the "master" branch */

git branch -d expert /* delete the "expert" branch – assuming there are no conflicts to fix or errors */

EXTENSION

Add another field to the Meeting table that is the date of the meeting description.