Rails: Views and Controllers

Lecture 28
Recall: Rails Architecture
Wiring Views and Controllers

- A controller is just an ordinary Ruby class
  - Extends ApplicationRecord
    ```ruby
class CourseRosterController < ApplicationRecord
  end
  ```
  - Location: app/controllers/
  - Filename: course_roster_controller.rb

- Actions are methods in that class
  ```ruby
def wake_up
  ... 
  end
  ```

- A view is an HTML page (kind of) that corresponds to that action
  - Location: app/views/course_roster/
  - Filename: wake_up.html.erb
  - Has access to instance variables (e.g., @student) of corresponding controller!
Recall: Rails Architecture

- **Model**
- **View**
- **Controller**
- **Dispatcher**
- **Routes**
- **Web Server**
- **Browser**

**GET /hi**

**app/**

**controllers/**

**course_roster_controller.rb**

**CourseRosterController**

#wake_up

**app/**

**views/**

**course_roster/**

**wake_up.html.erb**
Demo: Building From Scratch

$ rails new demoapp

Create CourseRosterController
  class CourseRosterController < ApplicationController
    create (empty) method wake_up
  Add route to config/routes.rb
    get 'hi', to: 'course_roster#wake_up'

Create wake_up.html.erb
  In app/views/course_roster

$ rails server
Example: Controller

#in app/controllers/
#filename course_roster_controller.rb

class CourseRosterController < ApplicationController
  def wake_up
    #for this simple example, no code needed
  end
end
Example: Route Definition

```
# in config/
# filename routes.rb

Demoapp::Application.routes.draw do
  get 'hi', to: 'course_roster#wake_up'

  # equivalent to (but shorter than):
  # match 'hi', to: 'course_roster#wake_up',
  #               via: [:get]
end
```
Example: View

<!--in app/views/course_roster/

filename wake_up.html.erb -->

<h1>Yo!!</h1>

<p>Are you awake?</p>
Single Point of Control

- Notice the duplication in names
- Controller name (**course_roster**) used in:
  - Name of the controller class
  - Filename of controller class implementation
  - Route
  - Directory containing view

- Action name (**wake_up**) used in:
  - Name of the method within controller class
  - Route
  - Filename of view source

- "Solution": generate all these parts

  ```bash
  $ rails g controller course_roster
  wake_up
  ```
Demo: Generating A Controller

$ rails generate controller prof ask_question visit_office

- Results in:
  - Modification to config/routes.rb
  - Creation of ProfController class
    app/controllers/prof_controller.rb
  - Definition of methods in ProfController
    def ask_question ... end
def visit_office ... end
  - Creation of 2 views (i.e. one per action)
    app/views/prof/ask_question.html.erb
    app/views/prof/visit_office.html.erb

$ rails server
Recall ERb: Embedded Ruby

- General templating mechanism
  - A string (usually contents of a file, "template")
  - Escaped bits of ruby
    - `<% code %>` execute ruby code ("scriplet")
    - `<%= expr %>` replace with result of expr
    - `<%# text %>` ignore (a comment)

- Example: a text file
  This is some text.
  `<% 5.times do %>`
  Current Time is `<%= Time.now %>`!
  `<% end %>`

- Process using erb tool to generate result
  `$ erb example.txt.erb > example.txt`

- Naming convention: `filename.outputlang.erb`
  - Example `index.html.erb`

- Many alternatives, eg HAML
Example: books/index.html.erb

```html
<h1>Listing Books</h1>
<table>
  <tr>
    <th>Title</th>
    <th>Summary</th>
    <th colspan="3"></th>
  </tr>
  <% @books.each do |book| %>
  <tr>
    <td><%= book.title %></td>
    <td><%= book.content %></td>
    <td><%= link_to 'Show', book %></td>
    <td><%= link_to 'Edit', edit_book_path(book) %></td>
    <td><%= link_to 'Destroy', book, :confirm => 'Are you sure?', :method => :delete %></td>
  </tr>
  <% end %>
</table>
<br />
<%= link_to 'New book', new_book_path %>
```
Recall Layout

- Returned HTML formed from: Layout + template
- Layout is the overall structure of the HTML page
- See: app/views/layouts/application.html.erb

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>… etc
</head>
<body>
  <%= yield =>
</body>

- Action's template replaces layout's yield
  - Template is action-specific content of response
- Layout is where you put site-wide styling
  - e.g., navigation bar, div's with CSS classes, footers
Defining and Choosing Layouts

- Default layout for responding to action in **ProfController**
  - app/views/layouts/prof.html.erb
  - If not found, then use app/views/layouts/application.html.erb

- Or controller can explicitly name layout
  ```ruby
class ProfController < ApplicationController
  layout "people/snazzy"
  #layout "people/snazzy", :except => [:show]
  ```

- There is an application-wide controller that can also specify a fall-back layout
  ```ruby
class ApplicationController < ActionController::Base
  layout "main"
  ```
Summary

- View/Controller coupling
  - Location of view from name of controller
  - Filename of view from name of action
  - Controller instance variables available
- ERb
  - Template for generating HTML
  - Scriplets and expressions
  - Other templating approaches exist (e.g., HAML)
- Layouts and templates