Web Application Architectures

Module 6: Presentation/User Interface
Lecture 5: Dynamic Content, Templates and Layouts

THE UNIVERSITY *of* NEW MEXICO

Controllers and Views

- The controller and view in the MVC design pattern are tightly coupled—controllers supply data to views, and controller actions are typically the targets of the links provided in views.

- In Rails, a controller makes every instance variable it creates available to the associated view files.

  Ex.

  - All post are retrieved in the `PostsController#index` method, and stored in the `@posts` array.
  - In the `app/views/index.html.erb` file this array is accessed using an iterator:

```
<% @posts.each do |post| %>
<tr>
    <td><%= post.title %></td>
    <td><%= post.body %></td>
    ...
</tr>
<% end %>
```

Dynamic HTML with ERb

- In Rails, dynamic content is generated using templates, and the most common templating framework is Embedded Ruby (ERb).

- ERb is a filter that takes a `.html.erb` template file as input and transforms it into an HTML output file as follows:

  - Normal HTML content is passed through the filter without modification.

  - Content between `<%=` and `%>` is interpreted as Ruby code and executed, with the results substituted back into the file as a string in place of the `<%= ... %>` string.

  - Content between `<%` and `%>` (no equal sign) is interpreted as Ruby code and executed, but the results are *not* substituted back into the output file.

Proper Use of ERb

- Although ERb allows you to insert Ruby code into your view, as a matter of good design, use it sparingly!
- E.g., in a RESTful architecture, the HTML code in ERb templates should specify the structure of the document, and the Ruby code should be used to provide (dynamic) information specific to particular resources.
- Application-level functionality and business logic should *never* be found in ERb templates.
- For generating HTML elements and formatting data in the view, there are numerous Rails helper methods – use them.

Layouts

- In order to generate the final HTML file that will be supplied to the browser, a layout file is invoked, passing the template to it as a block.
- By default the `app/views/layouts/application.html.erb` is used. This file is automatically created whenever you create a new Rails application.
- Advantage of layouts: By editing one file, and its associated stylesheet, we can change the look and feel of the entire site.
- If you want to have different layouts for the different parts of a site, create a layout file that has the same name as the controller you want to associate it with, and place it in the layouts folder.

  Ex. `app/views/layouts/posts.html.erb`

Layouts

The app/views/layouts/application.html.erb file (note the yield statement):

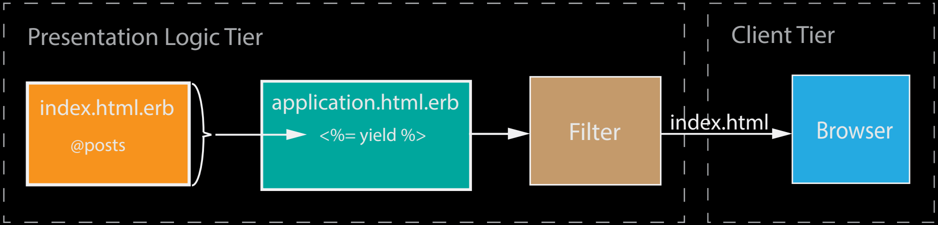
```
<!DOCTYPE html>
  <html>
  <head>
    <title>Blog</title>
    <%= stylesheet_link_tag "application", ... %>
    <%= javascript_include_tag "application", ... %>
    <%= csrf_meta_tags %>
  </head>
  <body>

  <%= yield %>

  </body>
  </html>
```

The HTML File

How the HTML file is created in response to an HTTP GET request for posts:



Helper Methods

There are numerous helper methods that are intended to be used in ERb templates. We saw a few in the `application.html.erb` file.

- The `stylesheet_link_tag()` helper method generates HTML <link> tags to the application's CSS stylesheets.
- The `javascript_include_tag()` does the same for the application's scripts.
- The `csrf_meta_tags()` method is included to prevent cross-site scripting attacks.