

IF140303-Web Application Development

Session-07: Introduction to the Phoenix Framework

PRU/SPMI/FR-BM-18/0222

Alfa Yohannis



Introduction to the Phoenix Framework

- Phoenix is a web development framework for building scalable and maintainable applications in Elixir.
- It is known for its performance, fault tolerance, and real-time capabilities.
- Phoenix requires installation of Phoenix itself, Node.js for asset management, and PostgreSQL for database management.

Why Install Node.js and PostgreSQL?

- **Node.js:** Handles assets like JavaScript and CSS, making it essential for the front-end part of the Phoenix application.
- **PostgreSQL:** A robust, open-source relational database system that Phoenix uses via Ecto for managing data persistence.

Phoenix Workflow Overview

- **Incoming Request:** Phoenix receives a request from the client.
- **Ensure HTML Request:** Phoenix checks if the request is for an HTML response.
- **Session Check:** Verifies if the request has an active session.
- **Security Check:** Performs necessary security validations.
- **HTTP Headers:** Adds appropriate headers for browser compatibility.
- **Access Request:** Determines what resource the request is trying to access.
- **Formulate Response:** Phoenix formulates the response and sends it back to the client.

Generating a Phoenix Project

- To create a new Phoenix project, use the command:

```
1  mix phx.new <name>
```

- Navigate to the project directory and create the database:

```
1  mix ecto.create
```

- Configure the database by editing `dev.exs`:

Database Configuration in dev.exs

```
1  config :reddit, Reddit.Repo,  
2    adapter: Ecto.Adapters.Postgres,  
3    username: "postgres",  
4    password: "postgres",  
5    database: "reddit_dev",  
6    hostname: "localhost",  
7    pool_size: 10
```

- Adjust the username and password according to your PostgreSQL setup.
- Start the Phoenix server with:

```
1  mix phx.server
```

Project Overview: Reddit-like Discussion Forum

- The project is a discussion forum similar to Reddit.
- Users can sign in with GitHub, view posts, create discussions, and comment.
- Users can also edit or delete their posts.

Server-Side Templating vs Single Page Apps (SPA)

- **Server-Side Templating:** The server generates HTML dynamically and sends it to the client.
- **Single Page Apps:** The client dynamically updates content without reloading the page, often using frameworks like React or Angular.
- Phoenix supports both approaches, allowing flexibility in web application design.

Phoenix Layout and MaterializeCSS PRADITA University

- Phoenix uses layout files to define the overall structure of HTML pages.
- MaterializeCSS is a modern front-end framework similar to Bootstrap, providing ready-to-use components and a responsive grid system.
- Add MaterializeCSS to your project by including its link tag in your layout file.
- Location at: web > templates > layout > app.html.eex

Implementing the Header with MaterializeCSS

```
1 <body>
2 <nav class="light-blue">
3 <div class="nav-wrapper container">
4 <a href="/" class="brand-logo">Logo</a>
5 <ul class="right">
6 <%= if @conn.assigns[:user] do %>
7 <li>
8 <%= link "Logout", to: session_path(@conn, :signout) %>
9 </li>
```

Implementing the Header with MaterializeCSS

```
1      <% else %>
2      <li>
3      <%= link "Sign in with Github", to: session_path(@conn, :
         request, "github") %>
4      </li>
5      <% end %>
6      </ul>
7      </div>
8      </nav>
```

Summary

- We introduced the Phoenix framework and discussed the necessary installations.
- Covered the Phoenix request-response workflow and project generation.
- Reviewed the project requirements, server-side templating vs. SPA, and implemented a MaterializeCSS header.