Introduction to Controllers in CodeIgniter

Topics : Codeigniter
Written on February 29, 2024

In CodeIgniter, controllers play a crucial role in handling user requests, processing data, and determining the appropriate response to be sent back to the user. Controllers act as intermediaries between the user's requests and the application's models and views. Here's an introduction to controllers in CodeIgniter:

Purpose of Controllers:

- Controllers in CodeIgniter are responsible for:
  - Receiving requests from the user's browser or client applications.
  - Interpreting and processing the incoming requests.
  - Invoking appropriate methods in the model layer to perform necessary actions, such as fetching data from a database or updating records.
  - Passing data to the view layer for rendering the response to the user.
  - Managing the flow of the application and directing the user to different pages or actions based on the request.

Creating Controllers:

- Controllers in CodeIgniter are typically stored in the `application/controllers` directory.
- Each controller is defined as a PHP class that extends the `CI_Controller` class provided by CodeIgniter.
- Controller class names should be in PascalCase, and the file name should match the class name with `.php` extension.
- Example of a basic controller:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class MyController extends CI_Controller {

    public function index() {
        // Controller method logic goes here
    }

    public function anotherMethod() {
        // Another controller method logic goes here
    }
}
?>
```

Handling Requests:

- CodeIgniter routes incoming requests to controllers based on the URL structure and routing configuration.
- By default, CodeIgniter routes requests to the `index()` method of the specified controller.
- Controllers can have multiple methods to handle different actions or pages within the application.
- Each method in a controller corresponds to a specific route or URL endpoint.

Loading Models and Views:

- Controllers can interact with models to perform database operations or retrieve data.
- Models are typically loaded within controller methods using CodeIgniter's built-in loader class.
- Similarly, controllers can load views to render HTML output or display data to the user.
- Views are usually loaded at the end of a controller method, and data is passed to views as an associative array.

Example:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

    public function index() {
        // Load model
        $this->load->model('welcome_model');

        // Call model method to retrieve data
        $data['message'] = $this->welcome_model->get_message();

        // Load view with data
        $this->load->view('welcome_message', $data);
    }
}
?>
```

In this example, the controller loads a model (`welcome_model`), calls a method to retrieve data (`get_message()`), and then loads a view (`welcome_message`) to display the data to the user.