

CodeIgniter - Handling form submissions

Topics : [Codeigniter](#)

Written on [March 01, 2024](#)

Handling form submissions in CodeIgniter involves validating form input, processing the submitted data, and responding appropriately based on the outcome of the form submission.

1. Create a View for the Form:

- Create a view file (form_view.php) containing the HTML form elements.
- Define form elements such as input fields, checkboxes, radio buttons, textareas, etc., inside the form.

Example form view file (form_view.php):

```
<!DOCTYPE html>
<html>
<head>
    <title>Form Submission</title>
</head>
<body>
    <h2>Submit Form</h2>
    <?php echo form_open('form/handle_submission'); ?>

    <label for="name">Name:</label>
    <input type="text" name="name" id="name" required>

    <label for="email">Email:</label>
    <input type="email" name="email" id="email" required>

    <input type="submit" value="Submit">

    <?php echo form_close(); ?>
</body>
</html>
```

2. Create a Controller Method to Load the Form:

- Create a controller method that loads the form view.
- Load the form view using the `$this->load->view()` method.

Example controller method:

```
public function show_form() { $this->load->view('form_view'); }
```

3. Create a Controller Method to Handle Form Submission:

- Create another controller method to handle the form submission.
- This method will process the submitted form data, validate it, and perform the necessary actions.

Example controller method to handle form submission:

```
public function handle_submission() {  
    // Load form validation library  
    $this->load->library('form_validation');  
  
    // Set validation rules  
    $this->form_validation->set_rules('name', 'Name', 'required');  
    $this->form_validation->set_rules('email', 'Email',  
'required|valid_email');  
  
    if ($this->form_validation->run() == FALSE) {  
        // Form validation failed, reload the form  
        $this->load->view('form_view');  
    } else {  
        // Form validation passed, process the form data  
        $name = $this->input->post('name');  
        $email = $this->input->post('email');  
  
        // Perform actions with form data (e.g., save to database, send  
email)  
  
        // Redirect or show success message  
    }  
}
```

4. Display Validation Errors (Optional):

- If form validation fails, you can display validation errors next to the form fields.
- Use `validation_errors()` function to display errors.

Example of displaying validation errors in the form view:

```
<?php echo validation_errors(); ?>
```