CodeIgniter - Performing CRUD operations using Models

Topics : Codeigniter
Written on February 29, 2024

Performing CRUD (Create, Read, Update, Delete) operations using models in CodeIgniter involves interacting with the database to manipulate data. Models encapsulate the logic for interacting with the database tables and abstract away the complexities of database operations. Here's how you can perform CRUD operations using models in CodeIgniter:

1. Creating Records (Create):

Controller:

```
public function create() {
    // Load the model
    $this->load->model('example_model');
    // Data to insert
    $data = array(
        'name' => 'John Doe',
        'email' => 'john@example.com',
        'age' => 30
    );
    // Call the model method to insert data
    $insert_id = $this->example_model->create_record($data);
    // Handle the result (e.g., redirect, display message)
}
```

Model:

```
class Example_model extends CI_Model {
    public function create_record($data) {
        $this->db->insert('users', $data);
        return $this->db->insert_id();
    }
}
```

2. Reading Records (Read):

Controller:

```
public function read() {
    // Load the model
    $this->load->model('example_model');
    // Call the model method to retrieve data
    $data['users'] = $this->example_model->get_records();
    // Pass data to the view for display
    $this->load->view('users_view', $data);
}
```

Model:

```
class Example_model extends CI_Model {
    public function get_records() {
        return $this->db->get('users')->result_array();
    }
}
```

3. Updating Records (Update):

Controller:

```
public function update($user_id) {
    // Load the model
    $this->load->model('example_model');
    // Data to update
    $data = array(
        'name' => 'Updated Name',
        'email' => 'updated@example.com',
        'age' => 35
    );
    // Call the model method to update data
    $this->example_model->update_record($user_id, $data);
    // Handle the result (e.g., redirect, display message)
}
```

Model:

```
class Example_model extends CI_Model {
    public function update_record($user_id, $data) {
        $this->db->where('id', $user_id);
        return $this->db->update('users', $data);
    }
}
```

4. Deleting Records (Delete):

Controller:

```
public function delete($user_id) {
    // Load the model
    $this->load->model('example_model');
    // Call the model method to delete data
    $this->example_model->delete_record($user_id);
    // Handle the result (e.g., redirect, display message)
}
```

Model:

```
class Example_model extends CI_Model {
    public function delete_record($user_id) {
        $this->db->where('id', $user_id);
        return $this->db->delete('users');
    }
}
```

Important Notes:

- Validate and sanitize user input to prevent SQL injection and other security vulnerabilities.
- Use transactions for atomic operations that involve multiple database queries to ensure data consistency.
- Implement proper error handling to handle database errors gracefully and provide meaningful feedback to users.