

CodeIgniter - Securing routes and controllers

Topics : [Codeigniter](#)

Written on [March 01, 2024](#)

Securing routes and controllers in CodeIgniter involves implementing mechanisms to control access to specific parts of your application based on user authentication and authorization. Here's how you can secure routes and controllers in CodeIgniter:

1. Authentication Middleware:

- Create a middleware function to check if the user is authenticated before accessing protected routes.
- Redirect unauthenticated users to the login page or display an error message.

Example Authentication Middleware:

```
function auth_middleware() {  
    $CI =& get_instance();  
    if (!$CI->session->userdata('logged_in')) {  
        redirect('login'); // Redirect unauthenticated users to the login  
page  
    }  
}
```

2. Apply Middleware to Routes:

- Apply the authentication middleware to routes that require authentication.
- You can do this by creating route groups and applying middleware to the group or by applying middleware directly to individual routes.

Example Route with Middleware:

```
$route['admin/dashboard'] = 'admin/dashboard';
```

3. Authorization Middleware:

- Create additional middleware functions to check if the authenticated user has the necessary roles or permissions to access certain routes or controllers.
- Redirect unauthorized users to a forbidden page or display an error message.

Example Authorization Middleware:

```
function admin_middleware() {
    $CI =& get_instance();
    if (!$CI->session->userdata('is_admin')) {
        show_error('You are not authorized to access this page', 403); //
        Display an error message for unauthorized users
    }
}
```

4. Apply Authorization Middleware:

- Apply the authorization middleware to routes or controllers that require specific roles or permissions.

5. Controller-Level Authorization:

- Implement authorization checks directly within your controllers' methods.
- You can use middleware functions or conditional statements to check the user's roles or permissions before allowing access to specific controller methods.

Example Controller Method with Authorization Check:

```
public function admin_dashboard() {
    if (!$this->session->userdata('is_admin')) {
        show_error('You are not authorized to access this page', 403); //
        Display an error message for unauthorized users
    }
    // Display admin dashboard
}
```

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)