ARYATECHNO

CodeIgniter - Using URI routing for clean URLs

Topics : Codeigniter
Written on February 29, 2024

URI routing in CodeIgniter allows you to define custom URL routes, making your URLs more descriptive and user-friendly. This feature enables you to map specific URLs to controllers and methods, providing cleaner and more intuitive URLs for your web application.

Here's how you can use URI routing in CodeIgniter:

1. Enable URI Routing:

URI routing is enabled by default in CodeIgniter. However, you can verify that it's enabled in your `application/config/routes.php` file.

```
$route['default_controller'] = 'welcome';
```

2. Define Custom Routes:

You can define custom routes in the same `routes.php` configuration file to map URLs to controllers and methods.

```
$route['about'] = 'pages/about';
$route['products'] = 'catalog/products';
$route['blog/(:num)'] = 'blog/post/$1';
```

3. Wildcard Routing:

You can use wildcard routing to match dynamic segments in URLs.

```
$route['products/(:any)'] = 'catalog/product/$1';
$route['category/(:num)/(:any)'] = 'catalog/category/$1/$2';
```

4. Regular Expression Routing:

For more advanced routing, you can use regular expressions to define complex URL patterns.

```
$route['product/([a-z]+)/(\d+)'] =
```

```
'catalog/product_lookup_by_category/$1/$2';
```

5. Route Parameters:

When defining routes, you can pass parameters to controller methods using the `(:any)`, `(:num)`, or custom regular expressions.

```
$route['user/profile/(:num)'] = 'user/profile/$1';
$route['blog/(:any)'] = 'blog/view/$1';
```

6. Catch-All Route:

You can define a catch-all route to handle requests that do not match any other routes.

```
$route['404_override'] = 'errors/not_found';
```

7. Route Groups:

You can group related routes together for better organization and readability. $route['admin'] = 'admin/dashboard';
$route['admin/users'] = 'admin/users';
$route['admin/posts'] = 'admin/posts';

Example:

Here's a practical example of using URI routing to create cleaner URLs:

```
$route['about'] = 'pages/about';
$route['products'] = 'catalog/products';
$route['product/(:any)'] = 'catalog/product/$1';
$route['category/(:num)/(:any)'] = 'catalog/category/$1/$2';
```

With these routes, URLs such as `/about`, `/products`, `/product/laptop`, and `/category/1/electronics` will map to their respective controllers and methods.