Bài 19: Rewrite URL trong Codeigniter

Trong bài này chúng ta sẽ tìm hiểu cách viết lại đường dẫn trong CI (Rewrite URL). Chức năng này khá quan trọng khi bạn làm web bởi vì nó thân thiện với người dùng và tốt cho SEO. Nếu gặp khách hàng hiểu biết SEO thì có khi họ yêu cầu bạn Rewrite URL theo ý muốn của họ.

Trong bài này mình hướng dẫn các vấn đề chính như sau:

- Tìm hiểu Route trong Codeigniter
- Rewrite URL cho trang sản phẩm
- Rewrite URL cho trang chi tiết
- Rewrite URL cho trang tag
- Rewrite URL cho trang chuyên mục

1. Tìm hiểu Route trong Codeigniter

Mọi thứ liên quan đến **Rewrite URL trong CI** đều nằm trong file application/config/routes.php, bạn mở file này lên nó sẽ có một số dữ liệu như sau:

```
$route['default_controller'] = 'welcome';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;<br><br>
```

Trong đó:

- **default_controller**: là controller mặc định, nếu bạn truy cập với URL ko có khai báo controller thì nó sẽ gọi tới controller mặc định này. Ví dụ: $route['default_controller'] = 'product/index';
- **404_override**: Nếu bạn không khai báo thì nó sẽ lấy trang mặc định lỗi của CI, còn nếu ban khai báo thì nó sẽ gọi đến controller và action trong đó. Ví dụ: $route['404_override'] = 'error/show404';
- **translate_uri_dashes**: Cái này không phải là route mà nó là một option với giá trị là true hoặc false. Nếu true thì CI sẽ chuyển đổi ký tự gạch ngang (-) thành gạch dưới (_) và ráp vào controller. Ví dụ bạn có controller tên Product_shop thì bạn sẽ gõ URL là http://domain.com/product-shop sẽ sai, nhưng nếu bạn khai báo $route['translate_uri_dashes'] = true; thì sẽ được.

Thêm mới route như thế nào?

Để thêm mới một route thì bạn chỉ cần thêm bên dưới cùng của file với cấu trúc:

```
$route['url-tren-trinh-duyet'] = 'controller/action/param1/param2...';
```

Để xử lý tốt **route trong CI** thì bạn phải biết khái niệm về biểu thức chính quy (regular expression). Ví dụ ta có chuỗi 'san-pham/([a-zA-Z0-9]+)/([0-9]+)' thì chúng ta có ba group:

- Group0: toàn bộ chuỗi 'san-pham/([a-zA-Z0-9]+)/([0-9]+)' ký hiệu $0
- Group1: ([a-zA-Z0-9]+) ký hiệu $1
- Group2: ([0-9]+) ký hiệu $2

Nghĩa là nếu ta khai báo cặp dấu () thì nó sẽ hiểu là một group. Tuy nhiên route trong **CI không sử dụng Group0**.

Các ký hiệu hay dùng trong Route

Route trong CI có các ký hiệu như sau:

- (:any) - đại diện cho các ký tự bất kì
- (:num) - đại diện cho các số tự nhiên

Sau đây là một số ví dụ sử dụng **routes trong codeigniter**:

```
$route['blog'] = 'blog/user';
```

Trong route này thì nếu trình duyệt là http://domain.com/blog thì nó sẽ gọi tới blog controller và user action.

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

Trong ví dụ này action user trong controller blog sẽ có một tham số truyền vào và giá trị của nó là đằng sau blog/.

- Nếu bạn gõ URL là http://domain.com/blog/thehalfheart thì sẽ **sai** vì ký tự đằng sau được khai báo là (:num)
- Nếu bạn gõ URL là http://domain.com/blog/12 thì **đúng**

Nhưng nếu bạn khai báo như sau thì cả hai URL trên đều đúng:

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

Sử dụng Regular Expression trong route

Bạn có thể sử dụng Regular Expression trong route. Thông thường chúng ta sử dụng các chuỗi regex sau:

- ([0-9]+) tương đương với (:num)
- ([a-zA-Z0-9]+) gần tương đương với (:any)

Ở đây chúng ta không liệt kê hết được, các bạn tự tìm hiểu về regular expression nhé.

Route Callback trong Codeigniter

Chức năng này mới được thêm từ Version CI3X, các versions trước không có.

Thay vì bạn khai báo là $route['url'] = 'value' thì bạn sẽ khai báo function cho nó với cú pháp như sau:

```
$route['url'] = function(){
    return 'controller/action';
};
```

Tới đây bạn sẽ có thắc mắc là nếu có tham số truyền vào controller thì sao? Đơn giản là tuân theo capturing group nhé. Nghĩa là **group1 tương đương tham số 1**, **group2 tương đương tham số 2**.

```
$route['product/(:any)-(:num)'] = function($any, $num){
    return 'controller/action/' . $any . '/' . $num;
};
```

Sau đây là phần áp dụng vào thực tế:

2. Rewrite UR cho trang sản phẩm

Trước tiên bạn tạo một controller Product như sau:

```
class Product extends CI_Controller{
   public function index($page = ''){
      echo '<h1>Home page</h1>';
      echo 'Page: ', $page;
   }

   public function category($cate_slug = '', $page = ''){
      echo '<h1>Category page</h1>';
      echo 'Slug: ', $cate_slug, '<br/>';
      echo 'Page: ', $page, '<br/>';
   }

   public function tag($tag_slug = '', $page = ''){
      echo '<h1>Tag page</h1>';
      echo 'Slug: ', $tag_slug, '<br/>';
      echo 'Page: ', $page, '<br/>';
   }

   public function detail($post_slug = '', $post_id = ''){
      echo '<h1>Detail page</h1>';
      echo 'Slug: ', $post_slug, '<br/>';
      echo 'ID: ', $post_id, '<br/>';
   }
}
```

Trong controller này, chúng ta tạo 4 action:

- **index**: trang chủ
- **category**: trang sản phẩm theo chuyên mục
- **tag**: trang sản phẩm theo tag
- **detail**: trang chi tiết sản phẩm

Rewrite trang chủ

Trang chủ ở đây không phải là trang home mà là trang hiển thị danh sách tất cả các sản phẩm nên nó sẽ có URL như sau:

- domain.com/san-pham
- domain.com/san-pham/page/1

Chúng ta sẽ viết hai routes như sau:

```
$route['san-pham/page/(:num)'] = 'product/index/$1';
$route['san-pham'] = 'product/index';
```

Tại sao mình lại đặt route có phân trang ở trên? Tại vì theo quy luật nó sẽ lặp danh sách các URL từ trên xuống dưới, nếu cái nào khớp thì nó sẽ ngưng. Chính vì vậy ta nên đặt có phân trang ở trên và ko có phân trang ở dưới. Bây giờ bạn chạy hai URL trên sẽ thấy kết quả.

Rewrite trang category

Trang danh sách sản phẩm có dạng sau:

- domain.com/chuyen-muc
- domain.com/chuyen-muc/page/1

Trong controller mình có nhận sẵn hai tham số rồi nên bây giờ ta chỉ viết routes thôi.

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

Các bạn để ý các ký hiệu $1 và $2 nhé, nó là **capturing group** đấy.

Rewrite trang tag

Tương tự như trang danh sách chuyên mục, nhưng để phân biệt giữa tag và chuyên mục thì trên URL mình sẽ thêm chữ tag nữa.

- domain.com/tag
- domain.com/tag/page/1

Và đây là routes:

```
$route['tag/page/(:num)'] = 'product/tag/$1/$2';
$route['tag'] = 'product/tag/$1';
```

Chúng ta cần đặt ruote này trên route của category, bởi vì trong route của category có $route['(:any)'] = 'product/category/$1'; nên nó sẽ đúng với bất kì trường hợp nào, chính vì vậy nó sẽ dừng và ko duyệt tiếp nên chúng ta phải đặt lên trên category.

```
// Home
$route['san-pham/page/(:num)'] = 'product/index/$1';
$route['san-pham'] = 'product/index';


// Tag
```

```
$route['tag/(:any)/page/(:num)'] = 'product/tag/$1/$2';
$route['tag/(:any)'] = 'product/tag/$1';

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

Rewrite trang detail

Trang chi tiết sản phẩm ở controller mình nhận hai tham số nên URL sẽ có dạng:

- domain.com/{tieu-de-cua-san-pham}-{id}.html

Chúng ta viết route như sau:

```
$route['(:any)-(:num)\.html'] = 'product/detail/$1/$2';
```

Tương tự bạn phải đặt route này trên cùng nhé, bởi vì nó là cái đặ biệt nên dễ bị trùng với category.

```
// Detail
$route['(:any)-(:num)\.html'] = 'product/detail/$1/$2';

// Home
$route['san-pham/page/(:num)'] = 'product/index/$1';
$route['san-pham'] = 'product/index';

// Tag
$route['tag/(:any)/page/(:num)'] = 'product/tag/$1/$2';
$route['tag/(:any)'] = 'product/tag/$1';

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

Vậy là bạn đã rewrite URL hoàn chỉnh.