

## Bài 2: Tạo Controller Trong CodeIgniter

Ở bài trước chúng ta đã tìm hiểu qua cấu trúc folder của CodeIgniter (CI), trong bài này, chúng ta bắt đầu tìm hiểu qua mô hình MVC trong CodeIgniter. Đầu tiên, chúng ta sẽ tìm hiểu về controller. Nội dung bao gồm:

- Tạo mới controller trong codeigniter
- Truyền biến vào controller
- Xác định controller mặc định
- Hàm khởi tạo
- Xóa đường dẫn index.php

### 1. Tạo Mới Controller Trong CodeIgniter

Tất cả các controller trong CodeIgniter đều được đặt trong thư mục **Application/Controllers** của CI. Mặc định khi cài đặt, CI đã tạo một controller tên là **welcome.php**, bạn xóa file này đi và tạo một file **hello.php** và điền nội dung vào là:

```
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index() {
        echo 'TKPro CodeIgniter xin chào các bạn';
    }
}
```

Dòng **if (!defined('BASEPATH')) exit('No direct script access allowed');** là dòng Security bảo vệ file của các bạn, nó không cho truy cập thẳng vào file mà phải thông qua file **index.php** ở mức ngoài cùng.

Lớp **Hello** là tên Controller của chúng ta, nó kế thừa lớp **CI\_Controller** của hệ thống CodeIgniter, tất cả các Controller đều phải **kế thừa** **CI\_Controller** thì mới sử dụng được các thư viện của CI và **tên Controller phải bắt đầu bằng chữ hoa**.

Hàm **index** là **Action** (method) của controller, đây là một hàm mặc định của tất cả các controller trong CodeIgniter nghĩa là nếu đường dẫn bạn chỉ gõ **domain.com/hello** thì mặc định nó sẽ chạy file **index**

Bạn ra trình duyệt gõ đường dẫn **localhost/ten\_project/index.php/hello/index**, kết quả xuất ra màn hình là dòng **"TKPro CodeIgniter xin chào các bạn"** thì chúng ta đã tạo mới thành công rồi.

Bạn vào file Controller **Hello** thêm một hàm other mới như sau:

```
class Hello extends CI_Controller {

    public function index() {
        echo 'TKPro CodeIgniter xin chào các bạn';
    }

    public function other(){
        echo 'TKPro CodeIgniter: Một hàm khác của controller Hello';
    }
}
```

Như vậy là ta đã tạo thêm một hàm (Action) mới trong controller Hello, bây giờ bạn ra trình duyệt gõ đường dẫn “localhost/codeigniter/index.php/hello/other” màn hình sẽ xuất hiện dòng “TKPro CodeIgniter: Một hàm khác của controller Hello”.

Qua hai ví dụ trên ta thấy **mỗi Controller trong Codeigniter** ta có thể tạo nhiều Action (hàm) trong đó, và mỗi action sẽ có những nhiệm vụ riêng biệt.

## 2. Truyền Biến Vào Controller

Trong mô hình MVC của các Framework, biến truyền vào theo phương thức GET đều có dạng “domain.com/controller/action/parameter1/parameter2/...”. Trong Codeigniter cũng vậy, để truyền biến vào Controller bạn sẽ có đường dẫn là “domain.com/index.php/controller/action/parameter1/parameter2/...”. Trong hàm (Action) của controller ta sẽ nhận nó bằng cách truyền những biến có vị trí tương ứng với từng parameter trên url.

### Ví dụ 1:

```
class Hello extends CI_Controller
{
    public function index($message = '')
    {
        echo 'TKPro CodeIgniter: ' . $message;
    }
}
```

Bạn ra trình duyệt gõ đường dẫn “localhost/codeigniter/index.php/hello/index/Hello Codeigniter Newbie” thì màn hình sẽ xuất hiện chữ “TKPro CodeIgniter: Hello Codeigniter Newbie”. Bạn hãy thử thay đổi biến truyền vào để kiểm tra.

Biến \$message truyền vào hàm (action) index mình gán nó giá trị khởi tạo bằng giá trị trống. Tại sao mình phải làm vậy? Tại vì theo nguyên tắc tất cả các hàm nếu truyền không đủ biến vào nó sẽ bị lỗi, nếu ta không gán giá trị mặc định thì nếu người dùng chỉ gõ “localhost/codeigniter/index.php/hello/index/” nó sẽ bị lỗi ngay, vì thế tất cả các biến truyền vào bạn phải gán giá trị mặc định để cho an toàn.

### Ví dụ 2:

```
class Hello extends CI_Controller
{
    public function index($id = 0, $message = '')
    {
        echo 'TKPro CodeIgniter: ID=' . $id . ', message = ' . $message;
    }
}
```

Trong ví dụ này mình truyền 2 biến vào hàm (action) index và gán giá trị mặc định cho biến, bây giờ bạn thử ra trình duyệt gõ các đường dẫn sau:

- localhost/codeigniter/index.php/hello/index/12/Hello => kết quả là “TKPro CodeIgniter: ID = 12, message = Hello”
- localhost/codeigniter/index.php/hello/index/12 => kết quả là “TKPro CodeIgniter: ID = 12, message = ”
- localhost/codeigniter/index.php/hello/index//Hello => kết quả là “TKPro CodeIgniter: ID =, message = Hello”

Qua ba ví dụ trên ta thấy các biến truyền vào hàm nó tuân theo thứ tự trên URL.

### 3. Xác Định Controller Mặc Định

**Controller mặc định** là controller sẽ được gọi khi trên url bạn không gọi đến một controller nào. Bạn vào file **application/config/routes.php** kéo xuống phía dưới tìm đến dòng `$route['default_controller'] = "welcome";`. Tại đây bạn sửa giá trị của biến `$route['default_controller']` thành tên controller mà bạn muốn chạy mặc định, ví dụ tôi sẽ sửa thành `$route['default_controller'] = "hello/other";` sau đó ra trình duyệt gõ đường dẫn **“localhost/codeigniter”** thì mặc định nó sẽ chạy đến controller **“hello/other”**.

### 4. Hàm Khởi Tạo

Trong lập trình hướng đối tượng thì tất cả các lớp đối tượng có hàm khởi tạo, hàm này sẽ chạy đầu tiên khi bạn khởi tạo một đối tượng mới. Trong PHP hàm khởi tạo được quy ước là đặt trùng tên với tên Lớp hoặc là bạn đặt tên **\_\_construct()**. Nếu trong Controller bạn muốn sử dụng hàm khởi tạo thì bắt buộc bạn phải gọi đến hàm khởi tạo của cha nó (CI\_Controller), vì trong **PHP** nếu hàm con kế thừa hàm cha mà hàm con có hàm khởi tạo thì nó sẽ chạy hàm khởi tạo của con chứ không chạy hàm khởi tạo của cha, mà trong hàm khởi tạo của cha lại chứa những đoạn code thiết lập hệ thống cho CI nên bắt buộc phải chạy nó.

```
class Hello extends CI_Controller
{
    // Hàm khởi tạo
    function __construct() {
```

```
// Gọi đến hàm khởi tạo của cha
parent::__construct();

}

public function index()
{
    echo 'TKPro.edu.vn';
}
}
```

### 5. Xóa Đường Dẫn Index.php trong codeigniter

Trong các ví dụ trên URL để gọi Controller trong codeigniter luôn có file index.php nhìn rất là mất thẩm mỹ, để bỏ file index.php trên đường dẫn url trong codeigniter bạn làm như sau:

Tạo file .htaccess cùng cấp với file index.php, tức là ở ngoài cùng, sau đó copy nội dung này vào:

```
Options +FollowSymlinks
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.+)/?$ index.php/$1 [L]
```

Sau đó bạn ra trình duyệt gõ URL “localhost/codeigniter/hello/other” thì CI tự động hiểu và gọi đến Controller **Hello** và hàm (Action) **other**.