

Bài 8: Tìm hiểu Helper Url và Form trong Codeigniter

Trong bài này, chúng ta sẽ tìm hiểu:

- Giới thiệu về helper
- Làm việc với form helper
- Các helper thông dụng

Codeigniter Framework tách biệt 2 khái niệm, đó là helper & library.

Helper là cái gì? Nó là những function được xây dựng sẵn nhằm hỗ trợ lập trình viên trong quá trình xây dựng website. Có rất là nhiều helper được khởi tạo sẵn.

Library khác helper ở điểm nào? Library được viết dựa trên những cái class dùng để định nghĩa các phương thức và thực thi các hành động ở bên trong lớp đó, dĩ nhiên nó khác biệt với helper. Helper là những thành phần mà chúng ta có thể gọi ở bất cứ nơi đâu và tùy ý sử dụng nó, ví dụ như gọi ra list menu ngay trong view chẳng hạn. Hai helper mà các bạn sẽ sử dụng nhiều nhất chính là **form & url**.

1. Load Helper

Cũng giống như library, để có thể sử dụng tất cả các hàm trong các helper thì việc đầu tiên là chúng ta phải gọi chúng ra. Chúng ta nên gọi helper ngay trong constructor của controller để có thể sử dụng helper đó ở mọi action. Có 2 cách để load helper: load lần lượt từng helper hoặc load một mảng các helper.

```
$this->load->helper('url');  
$this->load->helper(array('url', 'form'));
```

2. Url Helper

Helper này có rất là nhiều phương thức, tuy nhiên, kích thước của nó khá nặng, vì vậy chúng ta chỉ nên dùng một số phương thức phổ biến sau đây.

- *base_url()*: Chỉ định đường dẫn tuyệt đối, khai báo trong file config.php.
- *site_url()*: Nó gần giống với *base_url()*.
- *anchor()*: Khai báo một đường link, ví dụ *echo anchor('news/local/123', 'My News', 'title="News title"')*.
- *url_title()*: Hỗ trợ Url Friendly dạng *hom-nay-troi-co-mua*.
- *redirect()*: Hỗ trợ chuyển sang một trang được chỉ định.

Trong CI, chúng ta sẽ phải làm việc với router và khi phải làm việc với nó thì tất cả các đường dẫn sẽ được rewrite dưới dạng thư mục hoặc file, nếu chúng ta không chỉ định ra một đường dẫn tuyệt đối thì sẽ không thể nào định nghĩa được những đường dẫn

của thư mục images & css. Tránh lạm dụng phương thức trong helper vì làm như thế sẽ khiến ứng dụng website của các bạn chạy rất là chậm, bình thường chúng ta chỉ dùng 2 phương thức redirect & base_url để làm tốt công việc cấu hình đường dẫn tuyệt đối, chuyển trang khi thao tác xong một hành động nào đó.

3. Form Helper

Đây là một helper khá hay, nó giúp các bạn nhanh chóng tạo ra các form dùng để nhập liệu một cách nhanh chóng. Tuy nhiên, chúng ta cũng chỉ nên sử dụng nó tùy lúc thôi nhé, còn lại thì vẫn cứ dùng html mà tạo form bình thường.

Chúng ta sẽ tạo ra một controller tên là Form, gọi sử dụng helper url và form. Do chúng ta gọi một lần 2 helper nên chúng ta sử dụng dạng mảng. Chúng ta sẽ gọi các phương thức trong form helper ngay trong phần view. Trước khi bắt đầu ví dụ, chúng ta sẽ liệt kê những phương thức mà chúng ta sắp sử dụng.

```
<?php
class Form extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->helper(array('url', 'form'));
    }

    public function index(){
        $this->load->view("form");
    }
}
?>
```

- **form_open:** Mở thẻ form.
- **form_close:** Đóng thẻ form.
- **form_input:** Tạo thẻ input với type là text.
- **form_label:** Tương đương với thẻ label trong html.
- **form_password:** Thẻ input với type là password.
- **form_upload:** Thẻ input với type là file
- **form_textarea:** Thẻ textarea.
- **form_dropdown:** Thẻ select option trong form.
- **form_radio:** Thẻ tạo nút chọn trong form.
- **form_submit:** Thẻ input với type là submit.
- **form_fieldset:** Thẻ mở fieldset trong form.
- **form_fieldset_close:** Thẻ đóng fieldset trong form.

Có rất nhiều cách tạo ra form cũng như khai báo thông số cho nó, nhưng cách thường được sử dụng là tạo ra một mảng chứa các thông số. Ví dụ, chúng ta tạo ra một form input với name là username, value là tkpro.edu.vn thì trong CI, chúng ta viết như sau:

```
<?php
$user=array(
    "name" => "username",
    "size" => "25",
);
$pass=array(
    "name" => "pass",
    "size" => "25",
);
$email=array(
    "name" => "email",
    "size" => "25",
);
$gender1=array(
    "name" => "gender",
    "value" => "m",
    "checked" => TRUE,
);
$gender2=array(
    "name" => "gender",
    "value" => "f",
);
$opt=array(
    "1" => "Viet Nam",
    "2" => "Cambodia",
    "3" => "Malaysia",
);
$note=array(
    "name" => "note",
    "cols" => "40",
    "rows" => "5",
);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>freetuts.net</title>
</head>

<body>
<?php
    echo form_open(base_url()."index.php/form/index");
    echo form_fieldset("Member Register");
    echo form_label("Fullname: ").form_input($user)."<br />";
    echo form_label("Password: ").form_password($pass)."<br />";
```

```
        echo form_label("Email: ").form_input($email)."<br />";
        echo form_label("Gender:
    ").form_radio($gender1)."Male".form_radio($gender2)."Female<br />";
        echo form_label("Country: ").form_dropdown("Country: ", $opt, 1)."<br
    />";

        echo form_label("Note: ").form_textarea($note)."<br />";
        echo form_label(" ").form_submit("ok", "Register");
        echo form_fieldset_close();
        echo form_close();

    ?>
</body>
</html>
```

Như vậy, với các tạo ra các đối tượng trên form bằng cách sử dụng form helper và truyền mảng khai báo, chúng ta có thể định nghĩa nhiều thành phần cho form, mỗi thành phần có nhiều thuộc tính khác nhau như name, id, style, class, value,

4. Các Helper thông dụng

Mình sẽ liệt kê một số helper thông dụng thường được sử dụng trong CI. Có nhiều khuyến cáo là chúng ta sử dụng một cách hạn chế và không nên lạm dụng mà thay vào đó, chúng ta nên viết mã HTML thuần sẽ dễ kiểm soát và hiệu xuất làm việc của website cũng tốt hơn.

- CAPTCHA Helper
- Date Helper
- Download Helper
- Email Helper
- HTML Helper
- Language Helper
- Security Helper
- Text Helper
- String Helper