

Bài 10: Tìm hiểu library upload trong codeigniter

Trong bài này bạn sẽ được học:

- Giới thiệu về Library upload
- Thao tác & làm việc với nó

1. Giới thiệu library upload

Trong CI, với **Library upload** chúng ta sẽ làm được những việc sau:

- Hỗ trợ định dạng file được chỉ định khi upload.
- Tùy chỉnh độ rộng & cao tấm hình được phép upload một cách dễ dàng.
- Giới hạn dung lượng file được upload.
- Hỗ trợ chúng ta resize ảnh theo nhu cầu.
- Đóng dấu bản quyền watermark với định dạng text hoặc chèn logo lên bất kể vị trí nào trong tấm hình.

Với 5 ưu điểm trên, chúng ta thấy rằng chức năng upload file, đặc biệt là upload file ảnh, đáp ứng hầu hết các yêu cầu upload của một ứng dụng web.

2. Cấu hình library upload

Cũng như các library khác, để có thể thao tác với nó thì chúng ta phải gọi nó ra bằng cú pháp sau:

```
$this->load->library('upload')
```

Sau đó, chúng ta cần cung cấp một số thông tin cấu hình:

```
$config['upload_path'] = './uploads/';  
$config['allowed_types'] = 'gif|jpg|png';  
$config['max_size'] = '100';  
$config['max_width'] = '1024';  
$config['max_height'] = '768';
```

Với các thông tin trên thì các bạn có thể dễ dàng nhận ra rằng chúng ta đang cấu hình đường dẫn, định dạng hình ảnh được phép upload, kích thước tấm hình, cũng như dung lượng tấm hình được phép upload. Sau khi cấu hình xong các thông tin trên, chúng ta sẽ có câu lệnh để thực thi việc upload như sau.

```
$this->upload->do_upload('Ten file')
```

Và tất nhiên là sau khi upload xong thì chúng ta có thể kiểm tra xem mình upload có thành công hay không. Để lấy về thông tin của tập tin vừa upload, chúng ta sử dụng lệnh sau:

```
$this->upload->data();
```

Trong quá trình upload file nếu tấm hình của chúng ta bị lỗi cũng như có một số chuẩn mực không hợp lệ thì cần xuất ra thông báo lỗi để người dùng biết, để hiển thị thông báo lỗi, chúng ta sử dụng:

```
$this->upload->display_errors();
```

Như vậy thì chúng ta chỉ cần ghi nhớ 3 phương thức và một số thông tin cấu hình ở phía trên thì chúng ta hoàn thành chức năng upload hình ảnh hay tập tin nói chung.

3. Thực hành upload file từ máy tính lên web server

Trong ví dụ này chúng ta sử dụng helper form & url nên chúng ta sẽ load nó ra ngay tại constructor. Tạo thêm một folder uploads ngang cấp với folder application, chúng ta cũng tạo ra action index và load view có chứa form để upload file.

```
class Upload extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->helper(array("form", "url"));
    }

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

Tạo trong folder view một file có tên là upload_view có nội dung như sau.

```
<?php
$upload=array(
    "name" => "img",
    "size" => "25",
);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Freetuts.net</title>
</head>
<body>
    <?php
        echo form_open_multipart(base_url()."index.php/upload/doupload");
        echo form_label("Avatar: ").form_upload($upload)."<br />";
        echo form_label(" ").form_submit("ok", "Upload");
        echo form_close();
    ?>
</body>
</html>
```

Hàm **form_open_multipart** tạo ra form có thuộc tính *multipart/form-data* trong html, tạo đối tượng file (form_upload) với name là img, đối tượng nút bấm submit (form_submit) có name là ok. Form này có action trở đến controller là upload & action

sẽ là doupload. Thực hiện xong 2 thao tác này, sau khi mở trang web thì các bạn có thể view source để xem phần hiển thị html của nó.

Chúng ta sẽ tạo ra action doupload và check xem người dùng đã nhấn nút submit hay chưa bằng thao tác `$this->input->post('ok')`. Nếu nó thỏa điều này thì chúng ta mới tiến hành cho thông tin cấu hình upload file vào được. Việc check như thế này sẽ kiểm tra xem người dùng có chọn hình chưa, nếu chưa chọn hình mà nhấn nút submit thì trình duyệt sẽ trả về thông báo lỗi.

Khi cấu hình các thông tin xong, chúng ta sẽ load cái library và cung cấp thông tin cấu hình này (biến `$config`). Sau đó, chúng ta gọi method để thực thi thao tác upload file. Upload xong thì chúng ta hiển thị toàn bộ thông tin của tấm ảnh nếu việc upload là thành công. Ngược lại, nếu quá trình upload thất bại thì chúng ta sẽ xuất ra câu báo lỗi, khi đó chúng ta phải tạo ra một biến để đổ câu thông báo lỗi sang view (`$data['errors'] = $this->upload->display_errors();`). Vậy ở phương thức index, chúng ta phải khai báo biến `$data['errors']` bằng rỗng để nó ẩn câu thông báo lỗi, chỉ khi nào có lỗi thì mới hiển thị ra. Nếu errors khác rỗng thì chúng ta mới xuất thông báo lỗi. Bây giờ, chúng ta hãy trình duyệt và gõ <localhost/citest/index.php/upload> để kiểm tra.

```
public function index(){
    $data['errors'] = '';
    $this->load->view("upload_view", $data);
}

public function doupload(){
    if($this->input->post("ok")){
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size'] = '900';
        $config['max_width'] = '1024';
        $config['max_height'] = '768';
        $this->load->library("upload", $config);
        if($this->upload->do_upload("img")){
            echo 'Upload Ok';
            $check = $this->upload->data();
            echo "<pre>";
            print_r($check);
            echo "</pre>";
        }else{
            $data['errors'] = $this->upload->display_errors();
            $this->load->view("upload_view", $data);
        }
    }
}
```

Nếu upload thành công thì sẽ xuất ra các thông số như sau:

```
Upload Ok
Array
(
```

```
[file_name] => Koala.jpg
[file_type] => image/jpeg
[file_path] => D:/freetuts/www/cietest/uploads/
[full_path] => D:/freetuts/www/cietest/uploads/Koala.jpg
[raw_name] => Koala
[orig_name] => Koala.jpg
[client_name] => Koala.jpg
[file_ext] => .jpg
[file_size] => 762.53
[is_image] => 1
[image_width] => 1024
[image_height] => 768
[image_type] => jpeg
[image_size_str] => width="1024" height="768"
)
```

Để kiểm tra lỗi, chúng ta sẽ chọn thử một file .doc nào đó tiến hành nhấn nút submit xem kết quả.

The filetype you are attempting to upload is not allowed.

Kết quả là file này không đúng định dạng nên không được phép upload. Với việc cấu hình định dạng file cho phép upload thì các bạn đã phần nào hạn chế được việc hacker upload file shell (file thực thi) hoặc các mã độc khác lên website của mình.

4. Mã nguồn upload tập tin lên server

Controller Upload:

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

    public function index(){
        $data['errors'] = '';
        $this->load->view("upload_view", $data);
    }

    public function doupload(){
        if($this->input->post("ok")){
            $config['upload_path'] = './uploads/';
            $config['allowed_types'] = 'gif|jpg|png';
            $config['max_size'] = '900';
            $config['max_width'] = '1024';
            $config['max_height'] = '768';
            $this->load->library("upload", $config);
            if($this->upload->do_upload("img")){
```

```
        echo 'Upload Ok';
        $check = $this->upload->data();
        echo "<pre>";
        print_r($check);
        echo "</pre>";
    }else{
        $data['errors'] = $this->upload->display_errors();
        $this->load->view("upload_view", $data);
    }
}
}
```

View:

```
<?php
$upload=array(
    "name" => "img",
    "size" => "25",
);
?>
<!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
        if($errors != ""){
            echo $errors;
        }
        echo form_open_multipart(base_url()."index.php/upload/doupload");
        echo form_label("Avatar: ").form_upload($upload)."<br />";
        echo form_label(" ").form_submit("ok", "Upload");
        echo form_close();
    ?>
</body>
</html>
```