

## Bài 7: Phân trang trong Codeigniter

Trong bài này bạn sẽ được học:

- Các thao tác xử lý với phân trang.
- Các vấn đề mở rộng trong pagination.

Đây là một library cũng khá là phổ biến, hay được sử dụng trong quá trình xây dựng website, nó có nhiệm vụ giúp chúng ta phân trang một cách dễ dàng. Bản chất của library pagination là nó chỉ tạo ra những đường link, chứ nó không hỗ trợ chúng ta giới hạn kết quả trả về, để làm được điều đó chúng ta phải thao tác với model, tức là phải xử lý truy xuất database.

### 1. Load Library pagination

Cũng giống như các library trước, để có thể thao tác với nó, chúng ta phải load nó trong controller. Chúng ta tạo controller là Page và action là index. Trong controller Page chỉ có một action index nên chúng ta sẽ gọi library ngay trong action này. Trong constructor, chúng ta cũng phải load helper url ra để có thể khai báo đường dẫn đến controller và action muốn phân trang, helper là gì thì ở bài sau chúng ta sẽ tìm hiểu.

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

    public function index(){
        $this->load->library("pagination");
    }
}
```

Để bắt đầu tiến hành **phân trang** chúng ta cần phải khai báo một số tham số như sau:

```
$config['base_url'] = "Đường dẫn controller & action muốn phân trang";
$config['total_rows'] = "Tổng số record";
$config['per_page'] = "Số record trên một trang";
$this->pagination->initialize($config); // Chạy phân trang
```

Để giới hạn kết quả trả về chúng ta cần phải thao tác với model. Trong ví dụ này, chúng ta sẽ dùng lại file UserModel.php, cái đầu tiên mà chúng ta cần viết chính là phải tính tổng số record. Ở đây, chúng ta sử dụng hàm count\_all() dùng để tính tổng record.

```
public function countAll(){
    return $this->db->count_all($this->_table);
}
```

Tiếp theo, chúng ta sẽ **load model** ngay trong action index, thông tin đầu tiên mà chúng ta phải cấu hình chính là tổng số record trong database thông qua phương thức `countAll` mà chúng ta vừa viết trong model `UserModel`. Sau đó, cấu hình đường dẫn cho controller & action muốn phân trang, để nạp đường dẫn một cách chính xác, chúng ta sẽ dùng hàm **`base_url()`** được hỗ trợ trong helper url. Và tất nhiên chúng ta cũng phải cấu hình bao nhiêu record trong một trang, tức là phần **`per_page`**. Chúng ta phải truyền biến config vào library pagination để có thể thực thi việc phân trang. Cuối cùng, chúng ta in trực tiếp hàm **`create_links()`** trong controller để hiển thị kết quả phân trang, nên nhớ việc đổ dữ liệu chỉ nên thao tác trong view.

```
public function index(){
    $this->load->model("UserModel");
    $config['total_rows'] = $this->UserModel->countAll();
    $config['base_url'] = base_url()."index.php/page/index";
    $config['per_page'] = 3;

    $this->load->library('pagination', $config);
    echo $this->pagination->create_links();
}
```

Nếu màn hình kết quả trả về như sau xem như thao tác thành công.

1 2 3 > Last >

Để hiển thị dữ liệu thì chúng ta phải xử lý thêm vài thao tác trong model, ví dụ như dùng `result_array()` để lấy toàn bộ record và đổ nó ra một mảng, truyền tham số vào view & show dữ liệu ngay trong view. Tất nhiên là phải giới hạn kết quả, muốn giới hạn kết quả thì chúng ta dùng hàm `limit()` và truyền vào số record và vị trí bắt đầu.

```
public function getList($total, $start){
    $this->db->limit($total, $start);
    $query=$this->db->get("user");
    return $query->result_array();
}
```

Muốn hiển thị dữ liệu, chúng ta phải gọi phương thức `getList()` trong controller, truyền vào nó 2 tham số, số record trong một trang & vị trí bắt đầu, muốn có được vị trí bắt đầu thì ta phải get được đường dẫn (chỉ số) trang hiện hành. Trong CI, chúng ta có một helper hỗ trợ việc lấy đường dẫn và khuyến cáo không nên sử dụng phương thức `$_GET` vì nó không an toàn. Phương thức này là `$this->uri->segment(tham số)`. Để có thể lấy đường dẫn một cách chính xác, ví dụ link có dạng: `localhost/citest/index.php/page/index/???` thì `page` = controller ứng với vị trí số 1, `index` = action ở vị trí thứ 2, và tham số trang `$start` chính là vị trí thứ 3.

```
public function index(){
    $this->load->model("UserModel");
    $config['total_rows'] = $this->UserModel->countAll();
    $config['base_url'] = base_url()."index.php/page/index";
    $config['per_page'] = 3;
```

```
$start=($this->uri->segment(3)-1)*3;
$this->load->library('pagination', $config);
$data['data']= $this->UserModel->getList($config['per_page'], $start);
$this->load->view("page_view", $data);
}
```

Nhiệm vụ của `getList` là lấy dữ liệu về dưới dạng một mảng chứa thông tin user và chúng ta sẽ hiển thị trong view. Chúng ta tạo file `page_view.php` đơn giản có nội dung như sau:

```
echo "<pre>";
print_r($data);
echo "<pre>";
echo $this->pagination->create_links();
```

Nếu kết quả trả về như hình bên dưới thì chúng ta đã tiến hành phân trang thành công.

```
Array
(
    [0] => Array
        (
            [id] => 1
            [username] => admin
            [password] => 123456
            [level] => 2
        )
    [1] => Array
        (
            [id] => 2
            [username] => alibaba
            [password] => 123456
            [level] => 1
        )
    [2] => Array
        (
            [id] => 3
            [username] => vicky
            [password] => 123456
            [level] => 1
        )
)
1 2 3 > Last >
```

## 2. Các vấn đề mở rộng trong pagination

- Làm nổi bật link hiện hành, ví dụ: khi bạn click vào link 3 thì phía nó hiển thị như sau: 1 2 **3** 4 5.

```
$config['num_links'] = 2;
```

- Sử dụng link theo chỉ số trang:

```
$config['use_page_numbers'] = TRUE;
```