

## Bài 18: Tìm hiểu Library Shopping Cart trong CodeIgniter

Bài này sẽ giúp các bạn tìm hiểu một library khá quan trọng, đó là shopping cart. Vì thư viện Shopping Cart trong CodeIgniter được tích hợp sẵn Session nên các bạn phải đặt giá trị cho key encryption\_key trong file application/config.php nhé.

Trong CodeIgniter, Shopping Cart được lưu dưới dạng mảng và được mã hóa để lưu vào Session, mà Session trong codeigniter gắn liền với cookie (cookie lưu được 4Kb) nên bạn không thể lưu với dung lượng lớn được. Chính vì thế ta chỉ có thể lưu id sản phẩm và một số thông tin như giá cả chứ không thể lưu toàn bộ thông tin như hình ảnh, tiêu đề, tóm tắt.

### 1. Cấu hình library shopping cart

Muốn **thao tác với shopping cart trong CI** thì trước tiên phải load nó theo cú pháp:

```
<?php
class Shop extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("cart");
    }
}
```

### 2. Thêm sản phẩm

Shopping Cart trong CI lưu dưới dạng mảng và có những key bắt buộc không được sửa như id, qty, price, name. Ngoài ra nó có thêm key **options** chứa các thông tin con như size, color của sản phẩm nên nếu những key này không giống với database của bạn thì hãy xử lý trước khi insert để đồng bộ. Các bạn xem đoạn code dưới đây:

```
public function insert(){
    $data=array(
        "id" => "1",
        "name" => "Áo thun",
        "qty" => "1",
        "price" => "100000",
        "option" => array("color" => "màu đỏ"),
    );
    // Them san pham vao gio hang
    if($this->cart->insert($data))
        echo "Them san pham thanh cong";
    else
        echo "Them san pham that bai";
}
```

Trong đoạn code này tôi đã khởi tạo một sản phẩm mới chứa trong mảng **\$data**. Sau đó sử dụng hàm **insert** của **thư viện Shopping Cart** để tiến hành thêm sản phẩm vào giỏ hàng. Các bạn hãy chạy code và sẽ thấy kết quả thành công hay thất bại.

### 3. Hiện thị danh sách sản phẩm

Trong đoạn code hiện thị danh sách sản phẩm, chúng ta dùng hàm `contents` trong thư viện shopping cart để tiến hành xem toàn bộ thông tin sản phẩm.

```
public function show(){
    //Show thông tin chi tiết giỏ hàng
    $data=$this->cart->contents();
    echo "<pre>";
    print_r($data);
    echo "</pre>";
}
```

### 4. Xóa một sản phẩm

Muốn xóa một sản phẩm thì chúng ta phải biết được cái rowid của sản phẩm đó. Khi các bạn insert sản phẩm thì trường rowid tự động phát sinh, trường này có nhiệm vụ giúp chúng ta phân biệt từng sản phẩm một cách toàn diện.

Muốn có được giá trị **rowid** thì chúng ta bắt buộc phải sử dụng [vòng lặp foreach](#) để xuất dữ liệu. Và có thể xóa được một sản phẩm thì trước tiên chúng ta cần phải có hơn một sản phẩm trong giỏ hàng.

Chúng ta có đoạn code insert mới như sau.

```
public function insert(){
    $data = array(
        array(
            'id' => '1',
            'name' => 'Áo thun',
            'price' => '10000',
            'qty' => '1',
            'options' => array(color => 'màu đỏ')
        ),
        array(
            'id' => '2',
            'name' => 'Quần Jeans',
            'price' => '20000',
            'qty' => '1',
            'options' => array('color=> 'màu xanh')
        ),
        array(
            'id' => '3',
            'name' => 'Mũ lưỡi trai',
            'price' => '30000',
            'qty' => '1',
            'options' => array('color' => 'màu vàng')
        ),
    );
    // Them san pham vao gio hang
    if($this->cart->insert($data))
```

```
        echo "Them san pham thanh cong";
    else
        echo "Them san pham that bai";
}
```

### Kết quả:

```
Array
(
    [c4ca4238a0b923820dcc509a6f75849b] => Array
        (
            [rowid] => c4ca4238a0b923820dcc509a6f75849b
            [id] => 1
            [name] => Áo thun
            [qty] => 1
            [price] => 100000
            [option] => Array
                (
                    [color] => màu đỏ
                )
            [subtotal] => 100000
        )

    [72ea95e2650aca6c0707014d757b7ce4] => Array
        (
            [rowid] => 72ea95e2650aca6c0707014d757b7ce4
            [id] => 2
            [name] => Quần Jeans
            [price] => 20000
            [qty] => 1
            [options] => Array
                (
                    [color] => màu xanh
                )
            [subtotal] => 20000
        )

    [9acc658922ae3040b17a5ddaf35a4dbf] => Array
        (
            [rowid] => 9acc658922ae3040b17a5ddaf35a4dbf
            [id] => 3
            [name] => Mũ lưỡi trai
            [price] => 30000
            [qty] => 1
            [options] => Array
                (
                    [color] => màu vàng
                )
            [subtotal] => 30000
        )
)
```

Để xóa sản phẩm nào thì ta thiết lập qty = 0, ví dụ tôi muốn xóa sản phẩm có id là 1 thì việc trước tiên tôi phải lấy được rowid của sản phẩm đó bằng cách sử dụng **vòng lặp foreach** để lấy giá trị rowid. Sau đó trường rowid sẽ ứng với phần tử rowid mà chúng ta vừa lấy được.

Các bạn xem đoạn code xóa một sản phẩm.

```
public function deleteOne(){
    $data=$this->cart->contents();
    foreach($data as $item){
        if($item['id'] == "1"){
            $item['qty'] = 0;
            $del = array("rowid" => $item['rowid'], "qty" => $item['qty']);
        }
    }
    if($this->cart->update($del))
        echo "Xoa san pham thanh cong";
    else
        echo "Xoa san pham that bai";
}
```

## 5. Xóa hết sản phẩm

Hàm xóa hết sản phẩm của Shopping Cart trong CI giống với hàm hủy toàn bộ session. Chúng ta dùng hàm **destroy** để xóa hết toàn bộ sản phẩm tồn tại trong giỏ hàng.

```
public function del(){
    $this->cart->destroy();
    echo "Done";
}
```

## 6. Cập nhật sản phẩm

Giống như thao tác xóa một sản phẩm, để cập nhật một sản phẩm thì chúng ta cần phải có rowid của sản phẩm đó, rowid giúp chúng ta xác định sản phẩm cần cập nhật.

Để cập nhật sản phẩm có id là 2 thì chúng ta sẽ thiết lập qty cho nó lớn hơn 0, trường rowid sẽ nhận lấy giá trị phần tử rowid mà chúng ta vừa lấy được thông qua vòng lặp.

Các bạn xem đoạn code cập nhật sản phẩm.

```
public function update(){
    $data=$this->cart->contents();
    foreach($data as $item){
        if($item['id'] == "2"){
            $item['qty'] = 10;
            $update = array("rowid" => $item['rowid'], "qty" =>$item['qty']);
        }
    }
    if($this->cart->update($update))
        echo "Update san pham thanh cong";
    else echo "Update san pham that bai";
}
```

## 7. Các hàm khác

Hàm đếm tổng số sản phẩm trong giỏ hàng là hàm `total_items`.

```
public function total(){
    echo 'Hien co '.$this->cart->total_items().' san pham trong gio hang';
}
```

Hàm tính tổng số tiền có trong giỏ hàng là hàm `total`.

```
public function totalmoney(){
    echo 'Tong tien '.$this->cart->total().' $ trong gio hang';
}
```

Hàm `has_options(key)` dùng để kiểm tra rowid có tồn tại hay không, nếu tồn tại thì chúng ta mới lấy được các thông số ở bên trong. Hàm này thường được dùng chung với hàm `product_options(key)` có nhiệm vụ lấy ra các giá trị bên trong trường options. Cả 2 hàm đều phải thông qua vòng lặp foreach mới xuất dữ liệu được.

```
public function product(){
    $data=$this->cart->contents();
    foreach($data as $item){
        if($this->cart->has_options($item['rowid'])){
            foreach($this->cart->product_options($item['rowid']) as
$option_name => $option_value){
                echo "<b>$option_name</b>: $option_value<br />";
            }
        }
    }
}
```