

## ShopExV4.8 二次开发技术文档

### 1、 开发需求：

ShopExV4.8 网店系统是一套基于网上快速建店的标准化 B2C 电子商务系统。系统集成了最基本最普通最常用的电子商务运作流程及使用功能，可以满足正常的开店需求。

定制可以根据客户的需求对网站进行相应功能的添加修改或者删除，同时定制也存在一定的弊端。ShopExV4.8 以前版本的定制是在原来的程序上修改的所以定制过的网站就不能使用该版本后发布的相关补丁。

ShopExV4.8 版本采用 MVC 开发模式，二次开发解决了定制在原程序上进行修改导致程序不能升级的问题，使新的程序模块可以很好的融合到 ShopExV4.8 系统中同时也可以继承原有程序的所有功能。

### 二、 技术描述：

本着不与原程序冲突的原则，需要新建一个目录去存放二次开发所用的程序，这就要求在 ShopExV4.8 的配置文件中定义一个存放二次开发程序目录的常量。同时为了使二次开发程序能够兼容原程序的所有功能也要求要包含原来的控制器文件或模型层文件，通过类继承和函数重载的方式实现原有功能的保留、修改和新功能的开发，当然如果该功能完全与原有功能没有联系则只需继承控制器文件或模型层文件的基类。

### 3、 流程说明：

#### 1、 配置 config.php 文件：

`define('CUSTOM_CORE_DIR','自定义文件路径')` [自定义文件路径建议和 core 同级]

#### 2、 后台菜单新增规则(customSchema.php)：

格式参照原有后台菜单文件的书写格式但数组名必须为 \$cusmenu

##### ①、新增菜单项

此处格式参照 adminSchema.php 即可，数组名称注意应为 \$cusmenu

##### ②、在已有菜单项中添加

根据菜单出现的位置添加不同的参数

如在“统计报表”下新增二级菜单“测试二次开发”

```
$cusmenu['analytics']=array(  
    'items'=>array(  
        array(  
            'type'=>'group',  
            'label'=>'测试二次开发',  
            'position'=>'after|begin|end|before',  
            'reference'=>'访问统计',  
            'items'=>array(  
                array(  
                    'type'=>'menu',  
                    'label'=>'测试二次开发 1',  
                    'link'=>'index.php?ctl=vip/vote&act=index'  
                ),  
                array(  
                    'type'=>'menu',
```

```

        'label'=>'测试二次开发 2',
        'link'=>'index.php?ctl=vip/vote&act=index'
    )
    )
    )
    )
);

```

position 值及说明:

I、after: 在某个菜单项的后面, 此时 reference 必须为一个同级已存在的菜单项。如上述例子中为“访问统计”则“测试二次开发”出现在“访问统计”的下面

II、before: 同 I, 只是位置在前

III、begin: 在同级目录的最前面, reference 可以为空

IV、end: 在同级目录的最后面, reference 可以为空

③、在三级菜单中插入某菜单项同上述二级菜单的操作

### 3、控制器文件:

控制器文件必须以 cct 命名

①、继承原有控制器文件, 前提是原控制器文件必须存在

```

cct_原控制器名称 extends ctl_原控制名称{
    function 新增函数名(){
        //新增功能函数
    }
    function 原有函数名(){
        //函数重载
    }
}

```

②、新建控制器文件

/\*\*\*\*\*后台控制器文件\*\*\*\*\*/

```

cct_control extends adminPage{
    function __construct(){
        //自定义操作
    }
    //自定义函数
    .....
}

```

或者

```

cct_control extends objectPage{
    function __construct(){
        //自定义操作
    }
    //自定义函数
    .....
}

```

/\*\*\*\*\*前台控制器文件\*\*\*\*\*/

```

cct_control extends shopPage{
    function __construct(){
        //自定义操作
    }
    //自定义函数
    .....
}

```

#### 4、模型层文件：

模型层文件必须以 **cmd** 开头命名

- ①、继承原有模型层文件，前提是原模型层文件必须存在

```

cmd_原模型层名称 extends mdl_模型层名称{
    function 新增函数(){
        //新增函数
    }
    function 原函数名(){
        //重载函数
    }
}

```

- ②、新建模型层文件

```

cmd_model extends shopObject{
    function __construct(){
        //自定义操作
    }
    //自定义函数
    .....
}

```

或

```

cmd_model extends mdlFactory{
    function __construct(){
        //自定义操作
    }
    //自定义函数
    .....
}

```

#### 5、视图层文件：

视图层文件需要重新创建