

Wecenter 插件开发规范

1. 命名规范

插件命名以 `wc_` 作为前缀+插件名，如插件 `test`，即 `wc_test`

2. 目录新建

目录新建在根目录下 `plugin` 文件夹下，如插件名 `test`，目录结构就是 `/plugins/wc_test/`

3. 目录结构

目录包含内容必须要有配置文件、核心类，非必要的有模型类、静态资源、第三方类、页面模板等，如 `test`，则玩着目录如下

```
wc_test
|-config.php  配置文件
|-test.php    核心类
|-test_model  模型类
|-view        视图模板目录
|-static      静态资源目录
|-install.sql 安装 sql
|-uninstall.sql 卸载 sql
|-upgrade.sql 升级 sql
```

4. 配置文件

`Config.php` 配置文件 返回一个多维数组，包含有：

`name:` 插件标识

`title:` 插件标题名称

`intro:` 插件介绍

`author:` 插件作者

`version:` 插件版本

`state :` 插件状态

`config:` 也是一个数组，包含所需的配置项，如：

```
'config'=>[
  'ad_plugin_enable'=>
  array (
    'title' => '广告开启',
    'type' => 'radio',
    'value' => 'Y',
    'options' =>
    array (
      'Y' => '开启',
      'N' => '关闭',
    ),
  )
],
```

广告开启:

☒ 开启 ☐ 关闭

以上的配置 config 就是后台广告插件的配置显示效果, 'options' 对应的是配置项的值, 'value' 是默认值, 'type' 是配置类型, 常用的有 text, selectradio, CheckBox, 'title' 是配置名称, ad_plugin_enable 是开发者自己定义, 调用的时候调用此键名即可

```
<?php
return array (
    'config'=>[
        'ad_plugin_enable' =>
            array (
                'title' => '广告开启',
                'type' => 'radio',
                'value' => 'Y',
                'options' =>
                    array (
                        'Y' => '开启',
                        'N' => '关闭',
                    ),
            ),
    ],
    'name' => 'ad',
    'title' => '广告插件模块',
    'intro' => '实现前端广告添加控制',
    'author' => 'wecenter',
    'version' => '1.0.1',
    'state' => 2,
);
```

4. 核心类

核心类同样继承自 AWS_CONTROLLER, 包含 5 个固定方法, 即构造方法 install 安装, uninstall 卸载, enable 启用, disable 禁用, 构造方法获取插件的配置和插件相关信息, 安装、卸载、启、禁用四个方法默认是调用系统的方法, 也可以自定义, 在方法里自行处理相关逻辑。除了这 5 个方法之外, 开发者可以自定义其他的方法进行调用。

5. 模型类

模型类的定义名称需后接_model 方可, 内部的定义与使用与 model 目录下的模型相同,

```
if (!defined('IN_ANWSION'))
{
    die;
}

class ad_class extends AWS_MODEL{
    public function remove_pic($id){
        $info=$this->fetch_row('ad','id='.$id);
        @unlink('./uploads/'.$info['pic']);
        $this->delete('ad','id='.$id);
        // unlink($)
    }

    public function edit_ad($data, $adid = '')
    {
        return $this->update('ad', $data, 'id=' . $adid);
    }
}
```

另注意一点，此处的模型名不可与 `model` 目录下的模型重名，如果重名，会去调用 `model` 目录下的模型，从而会导致找不到相关的方法一类的错误。调用方法还是 `$this->model()` 进行调用。

6. static、view 静态资源、视图目录

Static 目录存放 `css\js` 等静态资源

View 视图目录可存放后台以及前台的视图模板，后台的要在目录下新建子目录 `admin`，前台的模板直接放在 `view` 下即可，在核心类中，通过 `PLUTPL::output('ad/banner');` 渲染页面，其中 `ad` 是插件名或者标识，`banner` 是页面名称，`PLUTPL::output('ad/admin/edit');` 后台的只需加上 `admin` 即可。

7. 安装、卸载、升级

安装、卸载、升级在后台都有相对应的操作，对应相应的 `sql` 执行即可。

8. 插件调用

正常的调用只需用公共方法 `hook()` 方法调用即可，在后台或者其他的页面需要调用插件方法进行页面调用或方法的时候，可以通过 `doact` 方法，前台 `explore/doact` 后台 `admin/plugin/doact` ,如

广告列表

添加广告

广告标题:

广告链接:

广告图片:

打开方式:

广告状态:

广告位置:

重置

确定

添加广告按钮需要调用插件添加广告的方法，此处是超链接，可以通过 `admin/plugin/doact/?p=ad&a=edit` 进行传参调用，其中 `p` 和 `a` 是必传的，`p` 是插件名，`a` 是方法名。