

ECSHOP

配送方式插件规范 v1.2

2006年9月21日

1.序言

该文档描述如何为 ECSHOP 创建一个配送方式插件以及配送方式插件的接口规范说明。

该文档的适用对象：

- 插件开发人员
- 技术爱好人员

1.1.文档历史

日期	版本	描述
2006-9-18	v1.0	第一个版本
2006-11-9	v1.1	类名修订为与文件名一致
2007-4-17	V1.2	增加对保价费用的支持

2. 内容

ECSHOP	1
配送方式插件规范 v1.2	1
1. 序言	2
1.1. 文档历史	2
2. 内容	3
3. 配送插件描述	4
3.1. 什么是配送方式?	4
3.2. 什么是配送插件?	4
4. 配送插件规范	4
4.1. 插件主体文件	4
4.1.1. 插件语言包	4
4.1.2. 插件注册信息	5
4.1.3. 配送方式插件类	6
4.2. 插件语言文件	7

3. 配送插件描述

3.1. 什么是配送方式？

配送方式是指客户在购买商品时所选择的货物运送的方式。每一个配送方式可以有多个配送费用计算标准相同的地区，我们称之为一个配送区域。每一个配送区域又有一个或多个关联的城市或地区。

客户在购买商品时，只有当客户所在的地区在某一配送方式的所辖地区之内的时候才允许客户选择该种配送方式。

3.2. 什么是配送插件？

配送插件用于指定配送费用的名称、描述、特性、费用的计算公式以及发货状态的查询等等。

4. 配送插件规范

配送插件包含的文件包括插件主体文件以及插件的语言文件。

4.1. 插件主体文件

插件主体文件存放在 `includes/modules/shipping` 目录下。插件主体文件包含两个部分：**插件主体信息**以及**插件类**。

4.1.1. 插件语言包

插件需要多语言支持时，需要包含自身的语言包。系统当前语言通过 `$GLOBALS['_CFG']['lang']` 定义。网站根目录通过 `ROOT_PATH` 常量定义。如下是一段包含自身语言包的片段。

```
$shipping_lang      =      ROOT_PATH.'languages/'.$GLOBALS['_CFG']['lang'].  
'/shipping/post_mail.php'; //语言包路径  
if (file_exists($shipping_lang))  
{  
    global $_LANG; //声明为全局变量  
    include_once($shipping_lang);  
}
```

4.1.2. 插件注册信息

插件注册信息包括了该配送插件的名称、版本等等。这些信息都保存在\$modules 数组中。以下是一个标准的注册信息范例：

```
if (isset($set_modules) && $set_modules == TRUE)
{
    $i = (isset($modules)) ? count($modules) : 0;

    /* 配送方式插件的代码必须和文件名保持一致 */
    $modules[$i]['code'] = basename(__FILE__, '.php');

    /* 插件的版本 */
    $modules[$i]['version'] = '1.0.0';

    /* 配送方式的描述 */
    $modules[$i]['desc'] = 'ems_express_desc';

    /* 配送方式是否支持货到付款 */
    $modules[$i]['cod'] = FALSE;

    /* 配送默认保价费用 */
    $modules[$i]['insure'] = '1%';

    /* 插件的作者 */
    $modules[$i]['author'] = "ECSHOP TEAM";

    /* 插件作者的官方网站 */
    $modules[$i]['website'] = "http://www.ecshop.com";

    /* 配送接口需要的参数 */
    $modules[$i]['configure'] = array(
        array('name' => 'base_fee', 'value'=>20),
        array('name' => 'step_fee', 'value'=>15),
    );
    return ;
}
```

注册信息说明：

- **code:** 配件插件的代码。**code** 的值必须和插件主体文件的文件名保持一致。同时 **code** 的值也是语言数组中的索引。通过在语言数组\$_LANG 中设置该索引的值来设定插件的名称。
- **version:** 插件的版本号
- **desc:** 插件的描述代码。**desc** 的值即是语言数组中的索引。通过在语言数组\$_LANG 中设置该索引的值来设定插件的描述内容。
- **cod:** 该配送方式是否支持“货到付款”这种支付方式。FALSE: 不支持; TRUE: 支持。
- **insure:** 配送默认保价费用, 可以是带百分比的字符串, 如'1%',表示收取商品总额的百分比; 也可以是具体数字, 如 10, 表示每次保价收取的费用, 为 0 时表示不支持保价。
- **author:** 插件作者姓名。
- **website:** 插件官方网站。
- **configure:** 配送接口参数可以定义多个参数。每个参数应包含两个索引 **name** 和 **value**。**name** 为参数代码, **value** 为参数默认值。

例如: 参数名称为基本费用。参数代码为'base_fee', 参数默认值为'20'。
那么我们做如下声明:

```
array('name' => 'base_fee', 'value'=>20)
```

并在插件语言文件中增加:

```
$_LANG['base_fee'] = '基本费用'
```

4.1.3. 配送方式插件类

类的命名

每一个配送方式插件类的名称必须和该插件的主体文件的文件名称保持一致。

以 **EMS** 这种配送方式为例

插件的主体的文件名为 **ems.php**, 则该插件类的名称就是 **ems**。如:

```
class ems
{
    function ems ($cfg=array())
    {
        ...
    }
    ...
}
```

同理, **邮件包裹** 主体文件名 **post_express.php**, 那么其类名应为 **post_express**。

类的属性

属性名称	类型	描述
configure	数组	保存插件的配送费用计算参数的数组

类的函数

1. 构造函数——与类名相同
 - 1) 参数: \$cfg
 - 2) 返回: void
 - 3) 描述: 构造函数用于初始化配送类的实例, 通过参数 \$cfg 来初始化配送类的 configure 属性。
2. 配送费用计算的函数——calculate()
 - 1) 参数: 无
 - 2) 返回: float
 - 3) 描述: 在配送费用计算的函数中我们需要指定一个配送费用的计算公式。同时查询当前客户订单中的商品重量、订单费用, 通过 **configure** 属性的值来计算出当前客户订单中商品所需支出的配送费用。
3. 查询发货状态的函数——query()
 - 1) 参数: 无
 - 2) 返回: string
 - 3) 描述: 如果该配送方式支持配送查询。可以在该函数中返回一个包含了查询表单(或链接地址的字符串)。
4. 计算保价费用函数——calculate_insure()
 - 1) 参数: 要保价商品总价\$total_price, 保价比例\$insure_rate
 - 2) 返回: float
 - 3) 描述: 此函数只有在保价费用为百分比时被调用。调用时外部将把要保价商品总价、保价比例传递进来。函数将返回计算后的保价费用。商品总价和保价比例都是 float 型。函数默认计算方法为 $\text{ceil}(\$total_price * \$insure)$ 。如果函数使用默认计算方法, 该函数可缺省不声明。

4.2. 插件语言文件

插件的语言文件存放于 languages/相应的语言目录/shipping 目录下。每一个语言文件的文件名必须和主体文件的文件名保持一致。

所有的语言项均包含在 **\$_LANG** 数组中。

语言文件中必须包含两项, 即插件名称以及插件的描述内容。这两项的索引就是在 4.1.2 插件注册信息所指定。