

【一】域名插件开发指南

如果开发一个域名插件，定义域名注册商标标识为demo,则结构如下

```
├─ Plugin
│   └─ DemoDomain [插件目录]
│       └─ Service [服务目录, 用于存放域名Api类]
│           └─ DomainApiDemo.class.php [核心类, 系统和API通信的所有实现]
│           └─ View [视图目录]
│               └─ config.html [系统设置参数的页面, 和config.php配合使用]
│               └─ DemoDomainPlugin.class.php [插件定义和实现的文件, 必须实现]
│               └─ config.php [配置文件, 一般用于配置密钥, 和config.html配合使用]
```

对于DemoDomainPlugin.class.php文件，需要修改的部分

`public $code = 'DemoDomain';` 修改为上面添加的code

`public $info = []` 修改为自己的插件信息

`public function domain_key(&$params)` 按照示例代码修改即可

config.php中的变量和view/config.html中的配置参数要一致，参数个数根据自己API接口添加或减少。

比如config.php包含`ak[]sk[]name[]dnsType`,view/config.html中包含`config[ak][]config[sk][]config[name][]config[dnsType]`

这里配置的参数通过setConfig方法传递给域名API对象

PluginDemoDomainServiceDomainApiDemo::setConfig[]如图

```
public function setConfig($config)
{
    $this->ak = $config['ak'];
    $this->sk = $config['sk'];
}
```

DomainApiDemo是系统和第三方API直接通信的，获取参数和实现的方法可参考DomainApiDemo.class.php

如果API接口不支持域名解析，而是在控制面板进行域名解析的，则不用实现DomainDNSRecordApi接口

```
class DomainApiDemo extends DomainApiBase implements DomainApi{}
```

不用实现以下这些方法

```
getDnsRecordList($argument)
getDnsRecordLines()
getDnsRecordTypes()
```

```
modifyAvailable()  
addDnsRecord($argument)  
addDnsRecord($argument)  
modifyDnsRecord($argument)  
deleteDnsRecord($argument)
```

如果需要通过API直接解析，需要实现DomainDNSRecordApi接口和上述方法

```
class DomainApiDemo extends DomainApiBase implements DomainApi, DomainDNSRecordApi
```

完成以上步骤，在 首页/系统/插件管理/插件列表/我开发的插件 中设置完参数，然后在 首页/生产/域名/域名后缀/域名后缀管理 设置默认注册接口为插件接口，则可以进行测试

如果测试没问题后可以打包上传审核。

插件打包并上传审核

登录 [阿帕云](#) 申请成为开发者，并在 软件-插件管理-开发者认证 申请开发者

插件标识应和插件目录名一致

插件包格式默认为zip, 打包时必须把整个目录一起打包

参考文档

可以参考示例代码 DemoDomain