iWebShop Api 接口开发文档

V2.7

目录

iWebShop Api 接口开发文档..... 1
Api 存在的意义.....2
Api 存放位置.....2
Api 开发说明.....2
应用范围.....4

Api 存在的意义

Iwebshop 商城系统中继承了强大的 api 接口功能，支持自主开发编写，不仅功能强大，复用性强，更可以把前端页面的设计人员从繁杂的 query 标签中解脱出来，更可以方便其他平台甚至是手机 app 进行调用。

Api 机制的存在极大的降低了开发模板的难度，可以让不懂任何程序的美工和前端设计人员轻松的获取到数据库中的数据，同时 api 接口也是程序封装的一个重要体现，更多复杂，通用型强的代码可以进行封装方便调用，开发人员只需要维护一份 api 定义即可，所以学习和掌握 api 是提高开发效率的一件利器！

Api 存放位置

接口的主体由两部分组成。

第一部分： /classes/api.php 这是 api 的核心，也是程序中调用的总入口，调用任何 api 都必须是通过此文件中定义的 Api 类来进行使用，例如 api 接口：

```
$data = Api::run('getArtList', 参数 1, 参数 2...);
```

这样一句代码就可以调用到商城系统的前 10 篇文章，怎么样？是不是很方便？

第二部分：/classes/api/api_resource.php 这里存放着 api 接口的配置和定义，每一条 api 都必须要在这个文件里面进行声明定以后才可以在系统中被调用到，这里算是 api 的一个总的路由，里面是以数组方式进行存储，如：

```
<?php
return array(
    'getGoodsList'     => array('file' => 'goods.php','class'=> 'APIGoods',),
    'getArtList'       => array('query' => array('name' => 'article','where' => 'vi
    'getRegimentList'  => array('query' => array('name' => 'regiment','where' => 'i
    'getPromotionList'=> array('query' => array('name' => 'promotion as p','join'
    'getCommendNew'    => array('query' => array('name' => 'commend_goods as co','j
    'getCommendPrice'  => array('query' => array('name' => 'commend_goods as co','j
    'getCommendHot'    => array('query' => array('name' => 'commend_goods as co','j
    'getCommendRecom'  => array('query' => array('name' => 'commend_goods as co','j
    'getOrderDistributed' => array('query' => array('name' => 'order','where' => '
);
```

我们看到所有的 api 名称都被列举出来，以数组 key 键的方式存在，然后在通过 Api 类进行调用。

Api 开发说明

我们有必要对 /classes/api/api_resource.php 中的编写规则进行说明，可以让用户自己编写出更强大的接口，目前 api 支持 2 种方式的编写，即：query 类型和 file 类型。配置类型在具体的 api 中写在数组 key 键中。

1．query 类型

对于熟悉 iwebshop 的开发者来说，query 并不陌生，就是利用 IQuery 类来获取数据库中的数据，在模板中就是{query:....}{/query} 而在 php 代码中就是 IQuery 类，这里 api 完全可以用 query 来获取到数据，用法和效果和模板标签 query 是保持一致的，比如：

```
'getArtList' => array('query' => array('name' => 'article','where' => 'visibility = 1 and top = 1','order' => 'sort ASC,id DESC','fields'=> 'title,id,style,color','limit' => '10'));
```

对于这么一条 api 来说，我们就可以理解为 query 类型，其中的

name：表名

where：条件子句

order：排序

fields：查询字段

limit：读取数量

这些属性名称与模板中的 query 标签都是保持一致的，一模一样的，这里只是把 query 标签进行二次封装到 api 里面，这样在模板中直接通过 Api::run('getArtList')；就可以获取到数据，也可以被反复利用，大大缩减了维护成本，而且缩减了 80%的字节量。

另外，query 类型也是可以有参数的，api 的参数支持追加到 Api::run('API 名称',参数...)默认情况下 query 参数有三种类型，array 数组（where 条件子句），int 整数（limit 读取数量），+ - 符号（order 正叙倒叙）。

Array 数组参数：直接对 api 中的 where 条件进行替换变量的操作。通常我们的 api 不一定是固定的常量，有时候条件是需要根据模板中的变量来生成的，还是用 getArtList 举例子，其中 where 条件是 `'where' => 'visibility = 1 and top = 1'` 我们可以把其中的任何内容变成可以通过传递参数进行改变的，`visibility=#visibility#` 其中的 `#visibility#` 就可以被 api 参数所替换掉，那么在使用的时候就是 `$data = Api::run('getArtList',array('#visibility#',0));` 这时候 Api 中的 `#visibility#` 就被替换成了 0 值，如果要替换多个字段就要增加多个参数。

Int 整数：读取 int 条数据，例如：`$data = Api::run('getArtList',array('#visibility#',0),20);`

+-符号：数据排序，例如：`$data = Api::run('getArtList',array('#visibility#',0),20,'+');`

另外需要说明的是，由于 api 的参数采用类型判断，所以参数的顺序没有要求，换句话说，`$data = Api::run('getArtList',20，'+',array('#visibility#',0));` 效果都是一样的。

2．file 类型

query 类型虽然可以解决绝对多数的查询问题但毕竟还是有局限性的，毕竟它仅仅是简单的从数据库中获取数据，没有任何逻辑和二次加工处理功能，比如我们需要开发一个数据比对的 api 接口，从获取数据中进行加工后返回结果；或者 api 并不一定非要与数据库交互，比如我们要循环遍历一些数据然后通过一些算法进行返回结果等，那么这类 api 用过简单的 query 肯定是无法实现的，那么这时候我们就需要 file 类型的 api 接口了。

file 类型的 api 把超高的自由度开放给了用户，用户可以自由编写自己的类，自由处理各种逻辑，实现各种需求。

```
'getGoodsList'   =>   array('file' => 'goods.php','class'=> 'APIGoods')
```

这个就是一个 file 类型的 api 接口，这个 getGoodsList 接口一共有 2 个参数，分别是 file

和 class。

file 参数：用户自己开发的 api 所存放的文件名称，注意这个文件必须存放到 /classes/api/ 这个目录下才可以被系统读取到。这个也是存放 api 接口的目录。

class 参数：要执行的类名称，也就是说，这个 file 文件中必须要有 class 这个类，并且这个类中必须要有 api 接口名字，比如：'getGoodsList' class 类中的 getGoodsList 就是调用的入口。

```php
class APIGoods
{
    /**
     * @brief  根据id获取商品信息
     * @param  int    $gid    商品id,多个id时以逗号','分割, 如：2,5,6,21
     * @return array  商品结果集
     */
    public function getGoodsList($gid)
    {
        $fields    = ' id , name , goods_no , sell_price , market_price, cost_price , store_nums , img , weight
        $dbObj     = IDBFactory::getDB();
        $tableName = IWeb::$app->config['DB']['tablePre'].'goods';
        return $dbObj->doSql('SELECT '.$fields.' FROM '.$tableName.' WHERE id in ('.$gid.') AND is_del = 0 ');
    }

}
```

只需要配置好参数以后，那么 api 被调用的时候就可以正常工作了，大体的工作流程就是内核找到 goods.php 这个文件，然后把 APIGoods 类进行实例化，最后再调用其中的 getGoodsList 方法。

当然，file 类型的接口参数依然是可以增加的，如图 getGoodsList 接口需要有 $gid 参数，那么我们在调用时候直接在 Api::run('getGoodsList','1,2,3'); 追加上即可。

应用范围

Api 接口大部分情况下用户 template 模板中，可以结合{foreach:}标签进行循环输出数据，也可以通过{set:}标签进行数据的暂时存储，使用方面比较灵活的，例如：

{foreach:items=Api::run('getGoodsList','1,2,3,4')}
{$item['name']}
{/foreach}
这样模板里面直接就可以输出 ID 为 1,2,3,4 商品的名称。

{set:$apiData = Api::run('getGoodsList','1,2,3,4')}
这样$apiData 变量中就存储着 ID 为 1,2,3,4 商品的名称。

总之 api 的使用层面要只要记住 Api::run('接口名称'); 就可以了！

我们在 iwebshop 系统中的 /classes/api/ 中提供了部分 query 和 file 类型的实例，用户自己自由翻阅查看，如有问题欢迎到 http://bbs.aircheng.com 进行咨询留言，我们期待您的参与。