**[C#开发微信门户及应用(22)-微信小店的开发和使用](http://www.cnblogs.com/wuhuacong/p/4203050.html)**

在做企业电子商务方面，微信小店虽然较淘宝天猫等起步较晚，但是作为一个电商平台，这个影响力不容忽视，结合微信的特点和便利，微信小店具有很好的粘合性和广泛的用户基础，因此花费一定的时间，在这方面做深入的研究和应用，也是我一个感兴趣的领域，本文基于前面微信系列文章的基础上，再对微信小店的内容进行一个系列化的介绍，希望对大家理解和使用上有所帮助，同时也把我自己的微信系列文章推向纵深的领域和方向。

**1、微信小店的申请和搭建**

微信小店的资质是必须为认证的公众号，并且需要认证后进行独立的申请，提交相关的企业信息和财务相关信息，手续和盖章的文件相对有点繁琐，不过为了好好研究和应用微信小店的功能，这些都没什么了，就按照他们的要求一步步提供相应的材料即可。

通过后，在自己的接口功能列表里面，可以看到对应的接口已经获取到了。

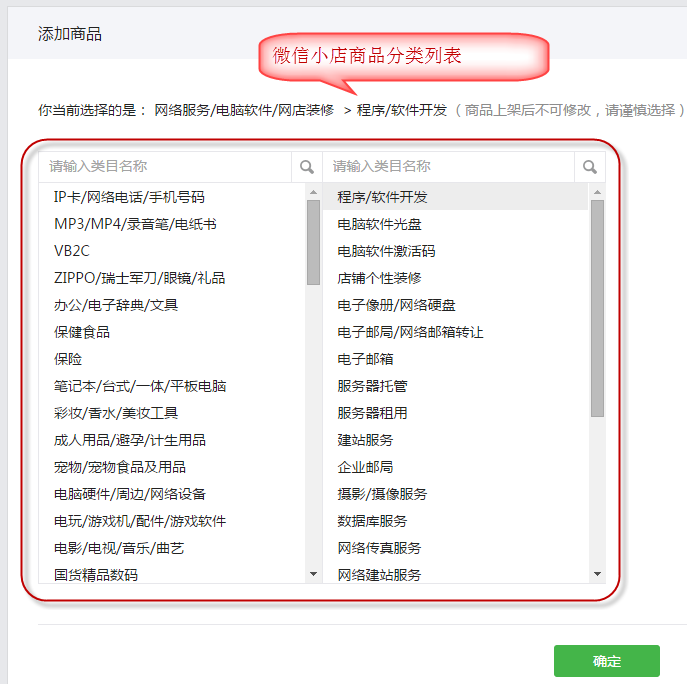


有了这些功能模块，第一步，我们就可以在微信公众号的管理平台上添加对应的商品信息，然后搭建好自己的微信店铺了。

我们进入微信小店，可以看到微信小店的响应功能操作界面。



我们第一步是需要添加对应的商品信息，从分类里面选择适合自己的分类，然后添加对应的商品信息和图片。



最后，我们添加完成了自己的商品列表（包括商品信息和商品分组的处理），完成后，类似的界面如下所示。



为了把商品合理的展示出来，微信小店引入了一个货架的概念，就是把商品分门别类的很好展示给客户，货架就是类似一个布局良好的展柜，我们可以定义不同的货架，然后公布不同的URL进行体验。



构建好货架信息后，我们就可以把货架的URL放到微信的菜单里面去了，这样我们就可以查看到自己的微信小店了。

当然店铺可以下单并进行处理了，客户下了订单，我们可以为微信小店的管理界面上进行订单的发货管理等操作。



在我们发货后，系统会有一个信息提示给对应购买者的微信号，如下所示。



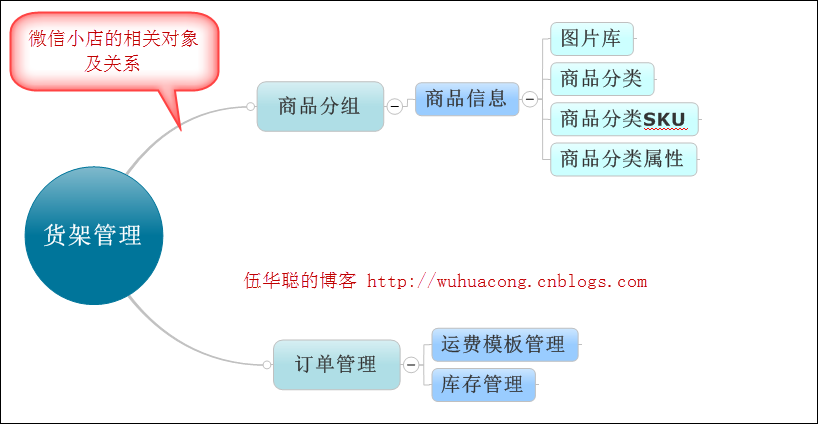
如果需要了解我的微信小店的功能，可以扫描下面二维码进行关注我们的认证公众号：**广州爱奇迪**



**2、微信小店的对象模型**

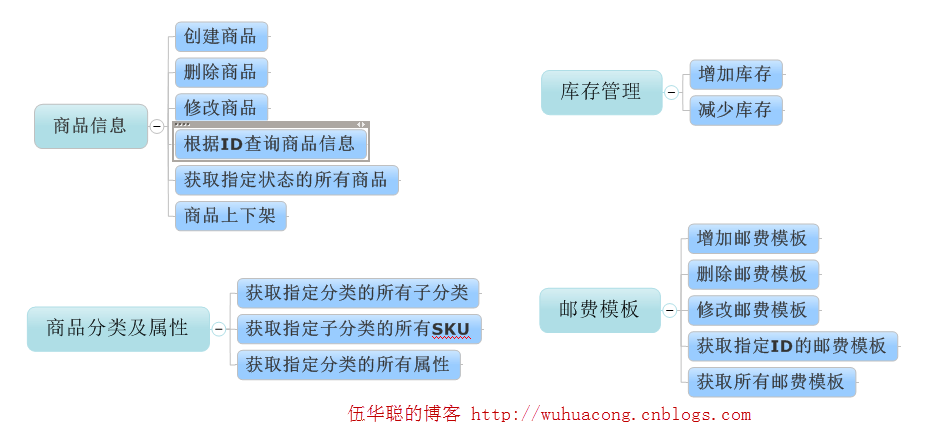
通过前面第1小节的学习，我们可能已经了解到了大概的微信小店的相关对象模型了，基本上就是包括了，常规的商品、商品分组、货架、库存、订单这些，还有商品分类，商品分类属性、商品分类SKU、快递邮寄模板、图片管理等功能。

为了更有效了解整个对象模型，我绘制了一些图形以帮助了解这些对象模型。

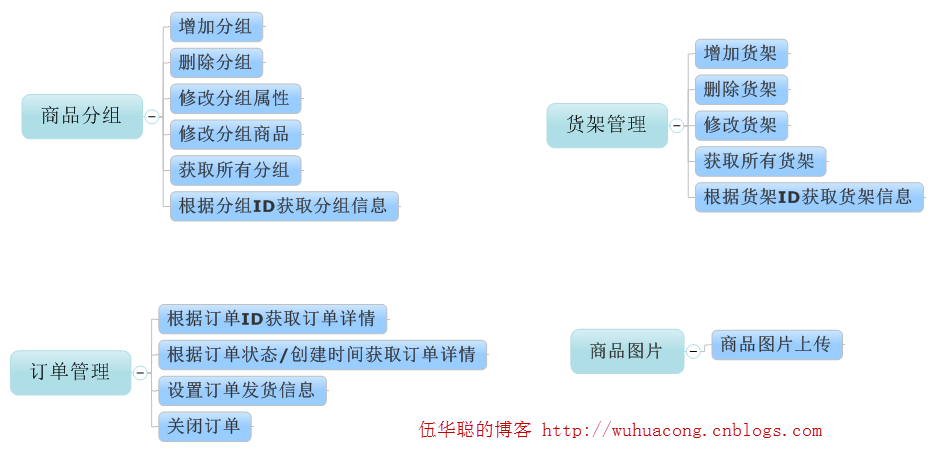


上面的图形，很好地阐述了这些对象的关系，基本上就是，我们第一接触的就是货架管理，通过货架入口，绑定了显示的商品分组和数量，我们就可以看到对应的商品信息，而商品通过图片、分类、属性、SKU等信息构建一个完整的商品对象，另外在货架的入口里面，我们可以对商品进行下订单，因此设计到订单和库存的管理，订单的发货需要有运费模板进行关联，从而实现完整的一个微信店铺的流程处理。

当然其中每个模型都有其对应的API接口，为了更加方便了解微信小店提供的功能接口，我通过图形列出上面涉及到的对象的功能接口，如下所示。



除了商品的对象模型接口，还有下面的一些接口。



**3、微信小店API的使用**

上面通过图示的方式，介绍了微信小店的相关对象和接口功能，这些我只是根据微信API提供的内容进行了汇总而已。

要了解更加详细的接口说明，我们还是需要参考微信的官方接口说明。

[微信公众平台开发者文档](http://mp.weixin.qq.com/wiki/static/assets/6f10c5d0769bf2d30c17e299736c0385.zip)

但是通过上面图示的接口功能图，我们可以整理出对应的微信小店的API接口和实现了。

下面的接口类和接口实现类，就是根据上面的分析和微信店铺的文档说明进行整理的C#开发封装处理。

由于接口涉及的内容比较多，因此我通过几个接口进行一定的见解，其他的如此类推。

例如对于微店商品的管理，我定义了下列的接口代码。

[复制代码](javascript:void(0);)

/// <summary>

///微小店管理的商品API接口

/// </summary>

public interface IMerchantApi

{

#region 商品信息

/// <summary>

/// 创建商品

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="merchantJson">商品对象</param>

AddMerchantResult AddMerchant(string accessToken, MerchantJson merchantJson);

/// <summary>

/// 删除商品

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="productId">商品ID</param>

CommonResult DeleteMerchant(string accessToken, string productId);

/// <summary>

/// 修改商品

/// product\_id表示要更新的商品的ID，其他字段说明请参考增加商品接口。

/// 从未上架的商品所有信息均可修改，否则商品的名称(name)、商品分类(category)、商品属性(property)这三个字段不可修改。

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="merchantJson">修改商品的信息</param>

CommonResult UpdateMerchant(string accessToken, MerchantJson merchantJson);

/// <summary>

/// 根据ID查询商品信息，如果成功返回MerchantJson信息，否则返回null

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="productId">商品的Id</param>

MerchantJson GetMerchant(string accessToken, string productId);

[复制代码](javascript:void(0);)

通过有效的封装处理，他们的一些实现代码也很简单，如下所示。

[复制代码](javascript:void(0);)

/// <summary>

/// 创建商品

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="merchantJson">商品对象</param>

/// <returns></returns>

public AddMerchantResult AddMerchant(string accessToken, MerchantJson merchantJson)

{

var url = string.Format("https://api.weixin.qq.com/merchant/create?access\_token={0}", accessToken);

string postData = merchantJson.ToJson();

return JsonHelper<AddMerchantResult>.ConvertJson(url, postData);

}

/// <summary>

/// 删除商品

/// </summary>

/// <param name="accessToken">调用接口凭证</param>

/// <param name="productId">商品ID</param>

/// <returns></returns>

public CommonResult DeleteMerchant(string accessToken, string productId)

{

var url = string.Format("https://api.weixin.qq.com/merchant/del?access\_token={0}", accessToken);

var data = new

{

product\_id = productId

};

string postData = data.ToJson();

return Helper.GetExecuteResult(url, postData);

}

[复制代码](javascript:void(0);)

基于文章篇幅的考虑，下面系列文章再进行单独的模型介绍和讲解。