

# 订单提交接口

接口说明：商家提供接口，当当调用。

地址：http://xxx.xxx.xxx/kaixin001/order.php

提交方式：POST

示例数据：

id=abcde&deal\_type\_name=code\_mine&kx\_order\_id=1234&user\_id=3127&user\_mobil  
e=13900000000&rebate=10&options=12:1,13:2&express\_addr=&express\_postcode=  
&express\_recver=&express\_phone=&express\_type=&express\_memo=&ctime=128486  
3557&sign=634bbe17eb3e2022a36cef530710e0f.....

参数说明：

红色部分，为必填项；蓝色部分，如果选择快递类商品，则必须填写。

id	团购单的唯一标识 id	主产品 ID
deal_type_name	业务类型，code_mine：团购网站提供的团 购券和其它网站提供的优惠代码、express： 需快递物品	是。 code_mine 或者 expressexpress;
kx_order_id	当当订单 ID	
user_id	用户 id（加密的字符串格式）	
user_mobile	用户手机号	
rebate	用户使用返利金额	
options	用户选择的商品信息(如果团购单要求用	子产品 ID

	户选择物品), 格式“子物品 ID1:购买数量 1, 物品 ID 2:购买数量 2”	
express_addr	邮寄地址	
express_postcode	邮编	
express_recver	收件人姓名	
express_phone	收件人电话	
express_type	邮寄投递方式(1-只在工作日配送(双休日、节假日不送), 2-只双休日、节假日送货(工作日不送), 3-学校地址, 白天没人, 请尽量安排其他时间送货(特别安排可能会超过预计天数), 4-任意时间均可送货)	
express_memo	用户配送相关特殊说明(用户自己写的信息)	
ctime	当当网调用接口的时间戳	
sign	校验串, 计算 sign 算法见附录 1	
express_fee	邮费	默认: 0
amount	单价*数量	单位(元), 精确到分
all_amount	单价*数量+运费	单位(元), 精确到分
pay_order_id	支付流水号	实际交易号
tcash	增加使用的账户余额	默认: 0
commission_used	增加使用的代金券	默认: 0

返回结果:

返回 xml 的数据，结果说明参考 order.xml

这个接口有可能重复提交（以 kx\_order\_id 为标识），合作方应该对每次提交都返回相同的数据，防止重复产生订单，和重复发券的行为

#### 返回数据样式:

```
<?xml version="1.0" encoding="utf-8"?>

<orders>

<order>

<id>bjbabyboss</id><!--团购单唯一标识 id，从 deal 获取的团购单 id-->

<order_id></order_id><!--合作网站订单 id-->

<kx_order_id></kx_order_id><!--当当网订单 id，提交订单的时候会提交过去-->

    <deal_type_name>code_mine</deal_type_name><!--业务类型-->

    <partner_result><!--合作方需返回给当当网的内容-->

    <code_mine>

        <sn>46946848</sn><!--团购网站自己的消费券号或者其它网站的电子优惠
码-->

        <code_type> 0</code_type><!--消费码类型。0：普通短信；1：二维码-->

        <verify>JTOMTO</verify><!--团购网站自己的优惠券密码，如果为其它网站的电
子优惠码没有密码则需返回空。-->

        <explain></explain><!--使用说明-->

        <expire_date>2010-09-18 15:00:00</expire_date><!-- 优惠券过期时间,字
符形式的北京时间-->

    </code_mine>
```

<pre>&lt;code_mine&gt;&lt;!--如果一个订单有多个码，只要重复 code_mine 即可--&gt;      &lt;sn&gt;46946848&lt;/sn&gt;      &lt;code_type&gt;0 &lt;/code_type&gt;      &lt;verify&gt;JTOMTO&lt;/verify&gt;      &lt;explain&gt;&lt;/explain&gt;      &lt;expire_date&gt;2010-09-18 15:00:00&lt;/expire_date&gt;  &lt;/code_mine&gt;  &lt;/partner_result&gt;  &lt;/order&gt;  &lt;/orders&gt;</pre>

## 附 1: Sign 值验证算法: c#

```
public static bool CheckSign()
{
    Dictionary<string, string> dic = new Dictionary<string, string>();

    // POST

    foreach (var p in HttpContext.Current.Request.Form.AllKeys)
    {
        dic.Add(p, HttpContext.Current.Request.Form[p]);
    }
}
```

```

// GET
foreach (var p in HttpContext.Current.Request.QueryString.AllKeys)
{
    dic.Add(p, HttpContext.Current.Request.QueryString[p]);
}

if (!dic.ContainsKey("sign"))
{
    return false;
}

string sign = dic["sign"];
dic.Remove("sign");
StringBuilder sb = new StringBuilder();
foreach (var p in dic.Keys.OrderBy(m => m))
{
    sb.Append(p).Append("=").Append(HttpUtility.UrlEncode(dic[p])).Append("&");
}

sb.Append("sn=" + Properties.Settings.Default.sn); //将为当当分配一串号码

string ns = sb.ToString();
foreach (Match m in Regex.Matches(ns, "[0-9a-f]{2}"))
{
    sb = sb.Replace(m.Value, m.Value.ToUpper());
}

string md5str =
System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(sb.ToString(),
"md5");

if (String.Compare(md5str, sign, true) != 0)
{
    return false;
}

return true;
}

```

## 附 2：错误代码

变量名	error code	desc
详细数据（格式为字符串）	0	成功
	1000	验证失败

	1001	验证失败
	1002	验证失败
	1003	验证失败：用户不存在
	1004	验证失败：没有数据节点
	2000	数据类型无法匹配
	2001	数据类型无法匹配
	2002	数据类型无法匹配
	3000	数据库无法写入
	3001	数据库无法写入
	3002	数据库无法写入
备注	code 包含失败描述以外的信息，合作方记录实际的 code 码，一旦产生故障，我方可以进行精确的故障排查。	