owitho开放平台接入指南

日期	版本	说明	作者
2018年5月19日	v0.1.0	初稿	钟杨杨
2018年5月20日	v0.1.1	添加了同步陈列接口, 增加了接口详细描述以及部分接口的补充描述, 更新了流程图添加了auth部分流程	罗迪
2018年5月22日	v0.1.2	更新了auth接口，增加utc字段	钟杨杨

接入说明

owitho开放平台是允许第三方使用点位服务的平台,owitho平台针对每个接入方平台账号会产生一个唯一的App Id,相对应App Id还会分配一个Secret Key,用于验证身份的合法性.第三方需要完成线下注册, 得到相应的App Id和Secret Key.

开发人员在接入实际业务接口前,要理解开放平台的签名和认证机制,业务接口以统一授权方式作为认证依据.

accessToken获取以及签名方法

accessToken获取:

* accessToken是第三方平台的全局唯一票据，第三方调用各接口时都需使用accessToken。第三方需要进行行妥善保存。
* accessToken的存储至少要保留512个字符空间。accessToken的有效期目前为24个小时，需定时刷新，重复获取将导致上次获取的accessToken失效。
* 开放平台以accessToken为接口调用凭据，来调用接口，所有接口的调用需要先获取accessToken，accessToken在24小时内有效，过期需要重新获取，但一天内获取的次数有限，开发者需要自自行行存储。

第三方平台API调用所需的accessToken的使用及生成方式说明:

1、为了保密secretKey，第三方需要一一个accessToken获取和刷新的中控服务器。而而其他业务逻辑服务器所使用的accessToken均来自于该中控服务器，不应该各自去刷新，否则会造成accessToken覆盖而影响业务；

2、目前accessToken的有效期通过返回的expire_time来传达，目前是24小时。中控服务器需要根据这个有效时间提前去刷新新accessToken。在刷新过程中，中控服务器对外输出的依然是老accessToken，此时开放平台后台会保证在刷新短时间内，新老accessToken都可用，这保证了第三方业务的平滑过渡；

3、accessToken的有效时间可能会在未来有调整，所以中控服务器不仅需要内部定时主动刷新，还需要提供被动刷新accessToken的接口，这样便于业务服务器在API调用获知accessToken已超时的情况下，可以触发accessToken的刷新流程。

4、第三方平台可以使用appId和secretKey调用 获取accessToken接口 来获取accessToken

1. 获取accessToken接口

接口调用请求说明:

```
 ${owitho.url}/getAccessToken?appId=4cdbc040657a4847b2667e31d9e2c3d9&salt=2345&utc=1526697000000&signature=9b7d70ab7f45d1256948811b91660c30
 ${owitho.url}需要更换为正确的地址，地址见附录
```

请求参数名	类型	必填	长度	说明	备注
appId	String	Y	-	第三方平台appId	
salt	int	Y	-	1000-9999内随机数	
utc	long	Y	-	请求时间戳	毫秒
signature	String	Y	-	签名	

签名计算规则:

(1) 将所有参数(signature除外)拼接(包括用户的secretKey)，并用&连接。所有接口的请求 方法GET/POST/PUT/DELETE的请求参数在计算signature之前需要进行utf-8编码

```
appId=4cdbc040657a4847b2667e31d9e2c3d9&salt=2345&utc=1526697000000&secretKey=fresh
```

(2) 将所得到的字符串用UTF-8进行urlEncode

```
appId%4cdbc040657a4847b2667e31d9e2c3d9%26salt%3D2345%26utc%3D1526697000000%26secretKey%3Dfresh
```

(3) 用(2)中字符串计算MD5，得到签名

```
9b7d70ab7f45d1256948811b91660c30
```

返回说明:

```
{
    "code":200,
    "msg":"success",
    "data":{
        "accessToken":"accessToken",
        "expireTime":1452570728594
    }
}
```

请求参数名	类型	必填	长度	说明	备注
code	int	Y	-	状态码	
msg	String	Y	-	返回信息	
accessToken	String	Y	-	获取到的凭证	
expireTime	long	Y	-	凭证到期时间（毫秒）	

错误时开放平台会返回错误码等信息，JSON数据包示例如下(该示例为AppID无效错误):

```
{
    "code":40001,
    "msg":"invalid appId",
    "data":{

    }
}
```

2.第三方平台请求时签名的计算规则:

第三方平台API调用必须严格按照如下数据格式:

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "data":"%7B%22name%22%3A%22jack%22%2C%22age%22%3A23%2C%22country%22%3A%22USA%22%7D",
    "salt":1234,
    "utc":1526697000000,
    "signature":"5fff5b80aeeef36e1c066b1004ec0262"
}
```

```
salt为1000-9999随机数
utc为请求发起的时间（毫秒单位时间戳）
```

签名计算规则:

(1) 首先获取请求的json字符串

```
{"name":"张三","age":23,"country":"中国"}
```

(2) 将上述字符串用UTF-8进行urlEncode，作为data的value值

```
%7B%22name%22%3A%22%E5%BC%A0%E4%B8%89%22%2C%22age%22%3A23%2C%22country%22%3A%22%E4%B8%AD%E5%9B%BD%22%7D
```

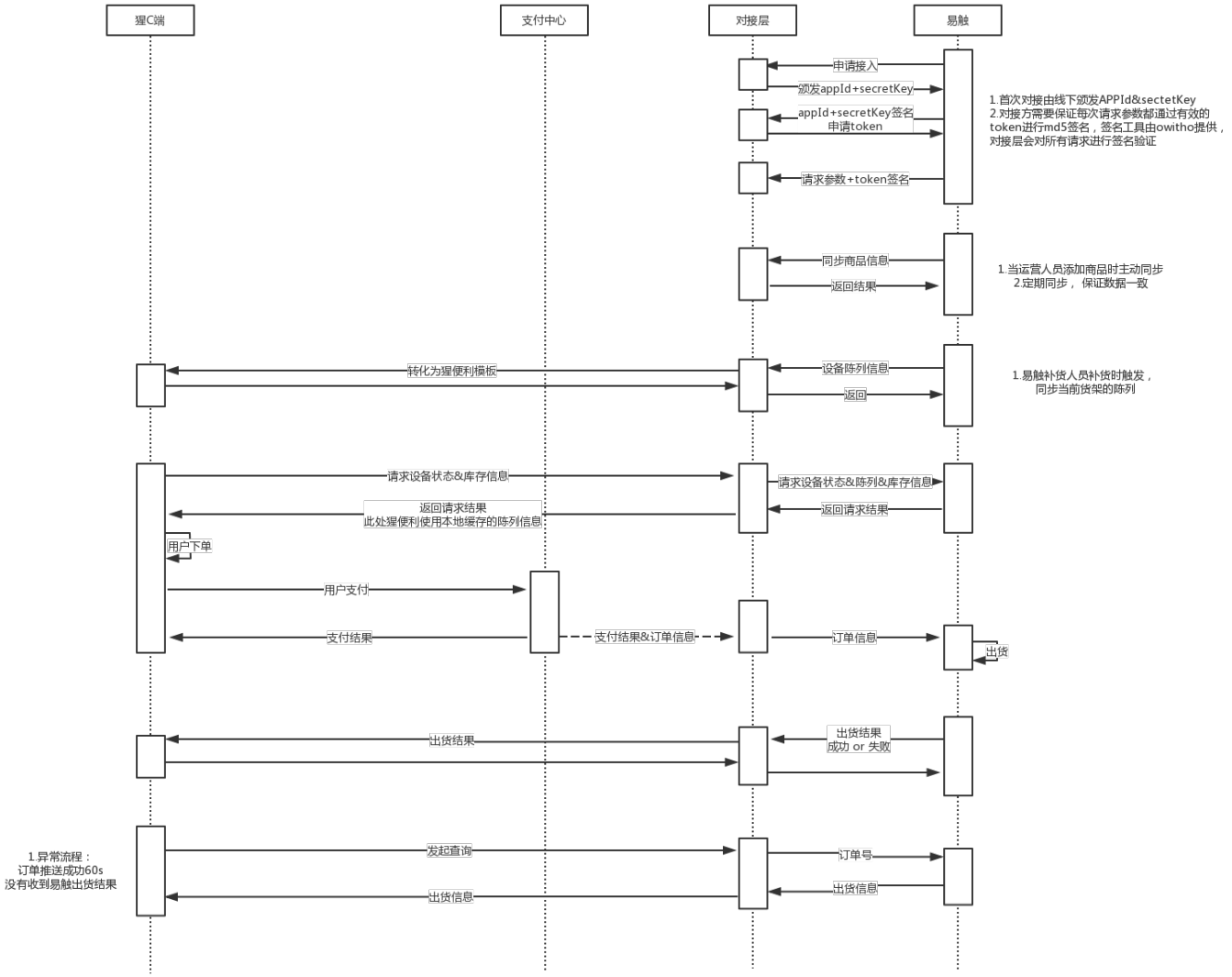
(3) appId, accessToken，以及上述字符串data和salt用&拼接起来

```
appId=4cdbc040657a4847b2667e31d9e2c3d9&accessToken=9e7d0604-cb0d-4c7c-a09d-cf365ce6936c&data=%7B%22name%22%3A%22%E5%BC%A0%E4%B8%89%22%2C%22age%22%3A23%2C%22country%22%3A%22%E4%B8%AD%E5%9B%BD%22%7D&salt=1234%26utc%3D1526697000000
```

(4) 用（3）中字符串计算MD5，得到签名

```
5fff5b80aeeef36e1c066b1004ec0262
```

业务流程



业务接口

以下所有业务接口，data中数据在请求中必须转为不带空格和换行符的字符串，然后用UTF-8进行urlEncode

1. 增量同步第三方商品信息接口

```
POST  ${owitho.url}/sku/sync
```

描述

当第三方接入owitho平台时， 需要第三方的商品信息,包括(商品编号,商品名称,商品图片等),以增量方式同步第三方商品信息到owitho平台

接口说明

请保证商品编号的唯一性,当商品编号重复同步时我们的策略是覆盖

API版本

```
0.1
```

调用方式

```
POST
```

请求参数

请求参数名	类型	必填	长度	说明	备注
skuCode	String	Y	-	商品编号	能够唯一表示商品信息的标识
skuName	String	Y	-	商品名称	
skuUnit	String	Y	-	商品单位	如：瓶、罐
skuSpec	String	Y	-	商品规格	如：100ml
originalPrice	String	Y	-	商品原始售价	
purchasePrice	String	N	-	商品进价	
skuPic	String	Y	-	公网可访问的商品图片url	

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":1234,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "data":{
        skuList:[
            {
                 "skuCode":"200000001",
                 "skuName":"可口可乐",
                 "skuUnit":"罐",
                 "skuSpec":"250ml",
                 "originalPrice":"10.50",
                 "purchasePrice":"5.50",
                 "skuPic":"https://img.xingbianli.cn/%08cola.png"
            },
            {
                 "skuCode":"200000037",
                 "skuName":"天喔茶庄蜂蜜柚子茶",
                 "skuUnit":"瓶",
                 "skuSpec":"180ml",
                 "originalPrice":"9.50",
                 "purchasePrice":"4.50",
                 "skuPic":"https://img.xingbianli.cn/%08cola.png"
            }
        ]
    }
}
```


返回参数说明

返回参数名	类型	说明	备注
code	int	状态码	
msg	String	返回信息	
data	String	返回数据	

返回结果示例

成功返回值

```
{
    "code":"200",
    "msg":"success",
    "data":{}
}
```

失败返回值 详见附录

```
{
    "code":"40000",
    "msg":"请求失败",
    "data":{}
}
```

2. 第三方补货时主动同步当前货架的陈列

```
POST  ${owitho.url}/machine/sync
```

描述

```
第三方补货人员在补货时触发,同步当前货架的陈列给owitho平台
```

接口说明

```
当存在合并货道时,开放平台不关心具体的合并逻辑,第三方只需要合并后的货道列表同步给开放平台即可.
示例:当前货道为1,2,3,4,5,6 此时将2,3货道合并后,同步的货道列表为1,2(合并后货道唯一标识),4,5,6
开放平台不关心2是货道2还是货道3
```

API版本

```
0.1
```

调用方式

```
POST
```

请求参数

请求参数名	类型	必填	说明	备注
machineCode	String	Y	机器code	
machineStatus	Int	Y	机器状态	10:正常，20:机器离线，30:机器机械故障详见附录
slotList	Array	Y	货道陈列详情	
slotNum	String	Y	货道号	
skuCode	String	Y	商品编码	
skuName	String	Y	商品名称	
skuDesc	String	N	商品描述	
originalPrice	String	Y	商品原始售价	
actualPrice	String	Y	商品真实售价	
maxInventory	Int	Y	最大库存数量	
currentInventory	Int	Y	实时库存数量	
slotStatus	Int	Y	货道状态	0:正常，其他:货道故障详见附录

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":1234,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "data":{
          "machineCode":"XHJSHnull1990000000108714",
          "machineStatus":10,
          "slotList":
             [
                 {
                     "slotNum":"01",
                     "skuCode":0000001,
                     "skuName":"600ml可口可乐",
                     "skuDesc":"600ml可口可乐",
                     "originalPrice":"10.50",
                     "actualPrice":"10.50",
                     "maxInventory":10,
                     "currentInventory":3,
                     "slotStatus":0
                 },
                 {
                     "slotNum":"02",
                     "skuCode":"0000002",
                     "skuName":"原味乐事薯片",
                     "skuDesc":"原味乐事薯片",
                     "originalPrice":"10.50",
                     "actualPrice":"10.50",
                     "maxInventory":10,
                     "currentInventory":5,
                     "slotStatus":0
                 }
             ]
    }
}
```


返回参数

返回参数名	类型	说明	备注
code	int	状态码	
msg	String	返回信息	
data	String	返回数据	

返回参数示例

```
{
    "code":"200",
    "msg":"success",
    "data"":{
    }
}
```

3. 根据机器code查询商品陈列、库存以及设备状态（第三方实现）

```
POST  ${third.url}/machine/query
```

描述

根据第三方机器code实时查询该机器的商品陈列、库存以及设备状态

接口说明

${third.url}为第三方平台的url，第三方需要在注册的时候提供

注意:当存在合并货道时,开放平台不关心具体的合并逻辑,第三方只需要合并后的货道列表同步给开放平台即可.
示例:当前货道为1,2,3,4,5,6 此时将2,3货道合并后,同步的货道列表为1,2(合并后货道唯一标识),4,5,6
开放平台不关心2是货道2还是货道3

API版本

```
0.1
```

调用方式

```
POST
```

返回参数

请求参数名	类型	必填	说明	备注
machineCode	String	Y	机器code	

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":1234,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "data":{
         "machineCode":"XHJSHnull1990000000108714"
    }
}
```

返回参数

请求参数名	类型	必填	说明	备注
machineCode	String	Y	机器code	
machineStatus	Int	Y	机器状态	10:正常，20:机器离线，30:机器机械故障详见附录
slotList	Array	Y	货道陈列详情	
slotNum	String	Y	货道号	
skuCode	String	Y	商品编码	
skuName	String	Y	商品名称	
skuDesc	String	N	商品描述	
originalPrice	String	Y	商品原始售价	商品以两位数点传输，如2.5转成"2.50"
actualPrice	String	Y	商品真实售价	商品以两位数点传输，如12转成"12.00"
maxInventory	Int	Y	最大库存数量	
currentInventory	Int	Y	实时库存数量	
slotStatus	Int	Y	货道状态	0:正常，其他:货道故障详见附录

返回参数示例

```
{
    "code":"200",
    "msg":"success",
    "data"":{
            "machineCode":"XHJSHnull1990000000108714",
            "machineStatus":10,
            "slotList":
                [
                    {
                        "slotNum":"01",
                        "skuCode":0000001,
                        "skuName":"600ml可口可乐",
                        "skuDesc":"600ml可口可乐",
                        "originalPrice":"10.50",
                        "actualPrice":"10.50",
                        "maxInventory":10,
                        "currentInventory":3,
                        "slotStatus":0
                    },
                    {
                        "slotNum":"02",
                        "skuCode":"0000002",
                        "skuName":"原味乐事薯片",
                        "skuDesc":"原味乐事薯片",
                        "originalPrice":"10.50",
                        "actualPrice":"10.50",
                        "maxInventory":10,
                        "currentInventory":5,
                        "slotStatus":0
                    }
                ]
    }
}
```


4. 订单通知接口（第三方实现）

```
POST  ${third.url}/order/notice
```

描述

用户下单成功后，把订单以及支付详情通知给第三方平台出货

接口说明

```
${third.url}为第三方平台的url，第三方需要在注册的时候提供
接收订单平台应做幂等处理，由于网络或者其他原因，没有接收到返回参数，相同订单有可能做重复推送
```

API版本

```
0.1
```

调用方式

```
POST
请求格式:application/json
```

请求参数

请求参数名	类型	必填	说明	备注
orderViewId	String	Y	订单号	接收订单平台应做幂等处理，相同订单有可能做重复推送
machineCode	String	Y	机器code	
discountMoney	String	Y	支付渠道优惠金额	
originAmount	String	Y	订单原始金额	
actualAmount	String	Y	订单实际金额	
payStatus	Int	Y	支付状态	0:待支付 1:支付成功 2:支付失败详见附录
refundStatus	Int	Y	退款状态	0:无退款 1:退款成功-全部 2:退款成功-部分 3:退款失败详见附录
orderTime	long	Y	下单时间	
payCompleteTime	long	Y	支付成功时间	
payMethod	Int	Y	支付方式	1:支付宝支付 2:微信支付 3:猩便利-余额支付 4:银行卡支付 5:现金支付 6:其他支付详见附录
orderDetailList	Array	Y	订单明细详情	
skuCode	String	Y	商品code	
quantity	Int	Y	商品数量	

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":1234,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "orderViewId":"XBL00000000000252",
    "machineCode":"XHJSHnull1990000000108714",
    "discountMoney":"12.00",
    "originAmount":"120.00",
    "actualAmount":"108.00",
    "payStatus":1,
    "refundStatus":0,
    "orderTime":1526696000000,
    "payCompleteTime":1526697000000,
    "payMethod":1,
    "orderDetailList":
        [

                {
                    "skuCode":"000000001",
                    "quantity":3
                },

                {
                    "skuCode":"000000002",
                    "quantity":5
                }
        ]
}
```

返回参数示例

```
{

    "code":"200",
    "msg":"success",
    "data"":""
}
```

5. 出货回调

```
POST  ${owitho.url}/order/callback
```

描述

第三方平台接收到出货订单后，回调owitho平台告知出货结果。

接口说明

出货结果按商品平铺展开，每一个商品对应一条出货详情记录。如：订单包含3瓶可乐、2包薯片，则出货详情对应5条记录。回调接口的出货状态不应该传100（出货中）。

API版本

```
0.1
```

调用方式

```
POST
请求格式:application/json
```

请求参数

请求参数名	类型	必填	长度	说明	备注
appId	int	Y	-	第三方平台ID	
salt	int	Y	-	1000-9999随机数	
signature	String	Y	-	签名	
utc	long	Y	-	请求时间戳	毫秒
orderViewId	String	Y	-	订单号	出货订单唯一标示
outStatus	int	Y	-	出货状态	200:出货成功 300:部分成功 400:出货失败 详见附录
outDetailList	Array	Y	-	出货详情	出货详情根据订单按单个商品平铺展开
slotNum	string	Y	-	货道号	
skuCode	string	Y	-	商品id	
status	int	Y	-	每个商品的出货状态	100:出货中 200:出货成功 300:部分成功 400:出货失败 详见附录

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":9527,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "data":{
        "orderViewId":"XBL00000000000252",
        "outStatus":200,
        "outDetailList":[
            {
                "slotNum":"01",
                "skuCode":"20000000100",
                "status":"01"
            },
            {
                "slotNum":"01",
                "skuCode":"20000000100",
                "status":"01"
            },
            {
                "slotNum":"02",
                "skuCode":"20000000200",
                "status":"01"
            },
            {
                "slotNum":"03",
                "skuCode":"20000000200",
                "status":"01"
            }
        ]
    }
}
```


返回参数

返回参数名	类型	说明	备注
code	int	状态码	
msg	String	返回信息	
data	String	返回数据	

返回结果示例

成功返回值

```
{
    "code":"200",
    "msg":"success",
    "data":""
}
```

失败返回值 详见附录

```
{
    "code":"40000",
    "msg":"请求失败",
    "data":""
}
```

6. 出货结果查询(第三方实现)

```
POST  ${third.url}/order/query
```

描述

```
第三方平台提供的订单出货结果查询接口，owitho在超时未接收到回调后，会主动查询出货结果
```

接口说明

```
根据唯一订单编号查询该订单的出货结果，出货结果按商品平铺展开，每一个商品对应一条出货详情记录。如：订单包含3瓶可乐、2包薯片，则出货详情对应5条记录。
```

API版本

```
0.1
```

调用方式

```
POST
请求格式:application/json
```

请求参数

请求参数名	类型	必填	长度	说明	备注
orderViewId	String	Y	-	订单号	

请求参数示例

```
{
    "appId":"4cdbc040657a4847b2667e31d9e2c3d9",
    "salt":9527,
    "signature":"5fsh8a99xddf5asqw90v12a",
    "utc":1526697000000,
    "data":{
        "orderViewId":"XBL00000000000252"
    }
}
```

返回参数

返回参数名	类型	说明	备注
code	int	状态码	
msg	String	返回信息	
data	String	返回数据	
orderViewId	String	订单号	出货订单唯一标示
outStatus	int	出货状态	100:出货中 200:出货成功 300:部分成功 400:出货失败 详见附录
outDetailList	Array	出货详情	出货详情根据订单按单个商品平铺展开
slotNum	string	货道号	
skuCode	string	商品id	
status	int	Y	-

返回结果示例

成功返回值

```
{
    "code":"200",
    "msg":"",
     "data":{
        "orderViewId":"XBL00000000000252",
        "outStatus":200,
        "outDetailList":[
            {
                "slotNum":"01",
                "skuCode":"20000000100",
                "status":"01"
            },
            {
                "slotNum":"01",
                "skuCode":"20000000100",
                "status":"01"
            },
            {
                "slotNum":"02",
                "skuCode":"20000000200",
                "status":"01"
            },
            {
                "slotNum":"03",
                "skuCode":"20000000200",
                "status":"01"
            }
        ]
    }
}
```


失败返回值 详见附录


```
{
    "code":"40000",
    "msg":"",
    "data":""
}
```


附录

通用返回码

状态码	说明
200	成功
40000	请求失败
40001	appid不存在
40002	验证签名失败
40003	salt不在范围内
40004	utc超时
50001	商品同步失败
50002	订单回调失败

状态字典

outStatus	说明
100	出货中
200	出货成功
300	部分成功
400	出货失败

机器状态

machineStatus	说明
10	正常
20	机器离线
30	机器机械故障

货道状态

slotStatus	说明
0	正常
其他	货道故障

支付状态

payStatus	说明
0	待支付
1	支付成功
2	支付失败

退款状态

refundStatus	说明
0	无退款
1	退款成功-全部
2	退款成功-部分
3	退款失败

支付方式

payMethod	说明
1	支付宝支付
2	微信支付
3	猩便利-余额支付
4	银行卡支付
5	现金支付
6	其他支付

owitho平台测试&生产地址

```
测试地址：https://open.owitho.intra.im/
生产地址：https://open.owitho.com/
```

GETTING START

fork me on github