

今通国际订单暂不能问题 解决接口 (OrderOpenApi)

文档版本信息				
日期	文档版本	接口版本	描述	编撰者
2016-10-12	V1.0	V1.0.0.0	初稿	今通国际研发中心

目录

目录	1
一.调用过程.....	2
1. 请求方式.....	2
2. 签名方式.....	2
3. 基本参数说明.....	2
3.1.请求基本参数.....	2
3.2.返回基本参数.....	3
3.3.标识代码.....	3
3.4.消息代码(MessageModel)	4
4. 执行过程.....	4
二.接口说明.....	5
1.暂不能出票问题解决(ProblemhasbeenSolve)	5
1.1.接口业务.....	5
1.2.参数说明.....	5
1.3.调用示例.....	6

一.调用过程

1. 请求方式

本接口仅支持 http 访问, 且仅支持 POST(FORM 表单 application/x-www-form-urlencoded) 方式传递参数.调用地址为 **ServerHostUrl**/Api/OrderOpenApi.ashx.

2. 签名方式

本接口中签名方式为 MD5 签名,签名按照**字母升序排序签名**,若相同首字母则看第二个字母,以此类推,排序后顺序把每个字段和值以"&"字符连接起来.

例如: 待签名参数: { "key1": "val1" , "key2": "val2" , "key0": "val0" }, 排序后待签名字符串:urlParam=" key0= val0&key1= val1&key2= val2";签名后字符为:Sign=Md5(urlParam+Key);若字段值为 **null** 或者**空字符串**,则**该字段和值不参与签名**.若对象中存在枚举值,那么签名的时候用的是枚举的.ToString()值,比如,枚举值 A=0 ,那么 A.ToString()="A",而非使用 int 值"0".若存在复杂对象引用那么对象字段也按照相同的方式进行签名.

例如: { "A": "aaa", "B": { "B1": "b111", "B2": "b222" }, "C": "c1" } 其中字段"B" 是指向的对象,那么"B"字段也按照上述方式签名.待签名字符串为:A=aaa&B=B1=b111&B2=b222&C=c1

本 Api 接口中,请求基本参数和返回参数只有 **Sign** 和 **SignType** 不需要签名,其他参数包括业务参数均需要签名.

注意签名时,不能出现循环引用,如果出现循环引用会导致程序内存溢出.利用 MD5 的签名函数对这个新的字符串进行签名运算, 从而得到 32 位签名结果字符串 (该字符串赋值于参数 sign)。

3. 基本参数说明

3.1.请求基本参数

参数类型	字段名称	类型	是否必填	是否需签名	描述
基本参数	Partner	string	是	是	合作 ID
	Action	string	是	是	业务动作
	RequestFmt	string	是	是	请求参数格式,只支持 XML
	ResponseFmt	string	是	是	返回参数格式,只支持 XML
	SignType	string	是	否	参数签名方式,只支持 MD5
	Sign	string	是	否	业务参数签名结果
	Version	string	是	是	业务动作版本
业务参数	Param	XmlNode[]	是	是	业务参数

3.2.返回基本参数

所有的返回结果又具有以下的字段

字段名称	类型	是否必填	是否需签名	描述
MessageModel	MessageModel	是	是	执行业务返回的结果信息,标识执行结果成功或失败的原因
Sign	string	是	否	返回参数签名结果
具体到业务方法更多的字段信息	XmlNodes[]	是	是	具体到业务方法更多的字段信息

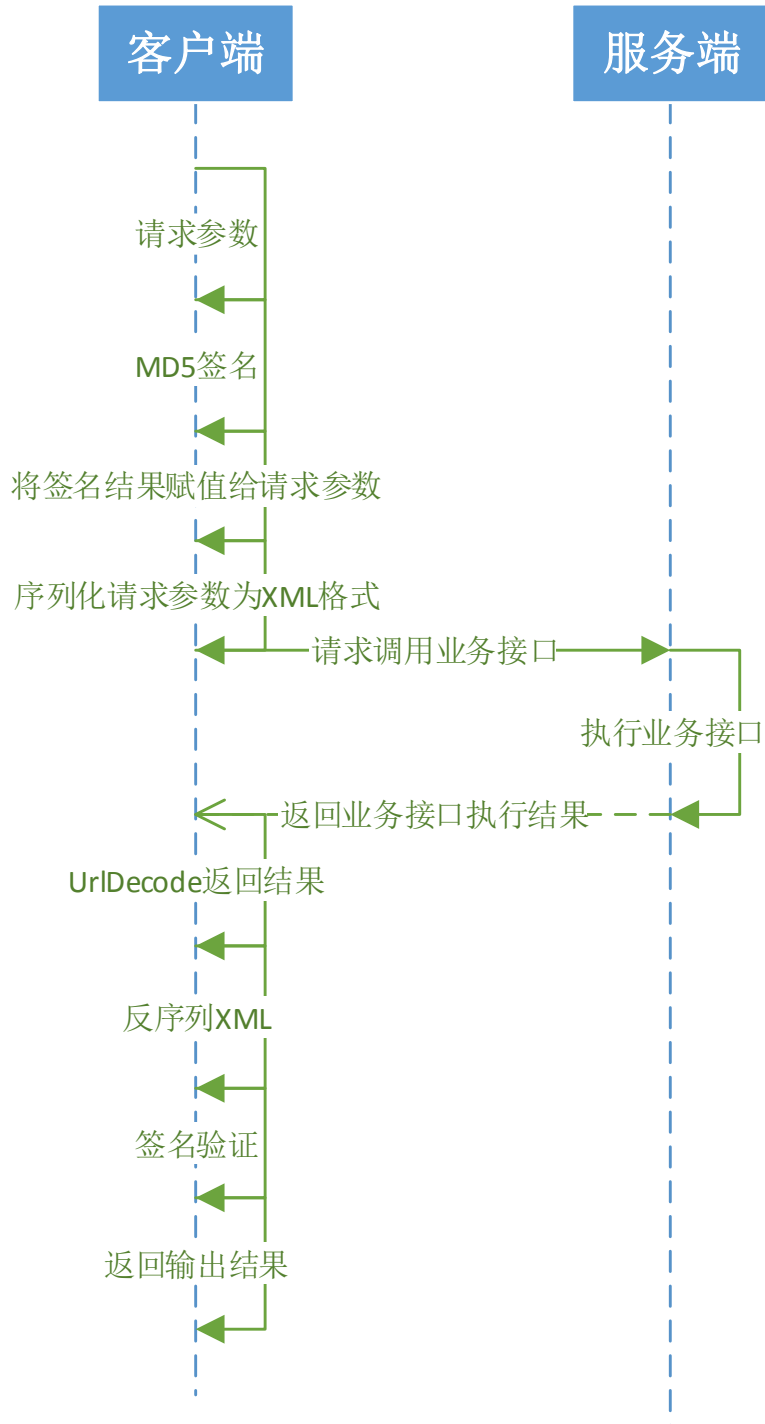
3.3.标识代码

标识代码	描述
Unknown	未知的
CallSuccess	调用接口成功
Authorised	有权限的
UnfoundUserInfo	未找到用户信息
UnauthorizedIp	无权访问的 IP
NotImplemented	该业务方法尚未实现
ArgumentIllegal	参数不合法
ArgumentDecryptFail	参数解密失败
ArgumentDeserializeFail	参数反序列化失败
SignFail	签名验证失败
CallFail	调用接口失败

3.4.消息代码(MessageModel)

类型	MessageModel	
字段名称	类型	描述
MessageCode	MessageCode	执行结果标识代码,详见 3.3.标识代码
Description	string	对标识代码的描述内容

4. 执行过程



二.接口说明

1.暂不能出票问题解决(ProblemhasbeenSolve)

1.1.接口业务

该接口把订单状态改成暂不能出票。.

1.2.参数说明

接口名称		ProblemhasbeenSolve		
请求参数				
字段名称	类型	是否必填	是否签名	描述
Partner	string	是	是	合作 ID
Action	string	是	是	业务动作
RequestFmt	string	是	是	请求参数格式,只支持 XML
ResponseFmt	string	是	是	返回参数格式,只支持 XML
SignType	string	是	否	参数签名方式,只支持 MD5
Sign	string	是	否	业务参数签名结果
Version	string	是	是	业务动作版本
Param	CannotReasonSolveParam	是	是	业务参数
业务参数: CannotReasonSolveParam				
字段名称	类型	是否必填	描述	
OrderID	string	是	订单号	
Remark	string	是	暂不能问题已解决处理备注	
输出参数				
字段名称	类型	是否必填	是否签名	描述
MessageCode	MessageCode	是	是	执行业务返回的结果信息,标识执行结果成功或失败的原因,见一.3.4
OrderID	string	是	是	订单号
Sign	string	是	否	返回参数签名结果

1.3.调用示例

1.3.1.请求参数

一个可能的请求参数:

```
<Request
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Partner>jinriguoji1</Partner>
  <Action>ProblemhasbeenSolve</Action>
  <RequestFmt>XML</RequestFmt>
  <ResponseFmt>XML</ResponseFmt>
  <SignType>MD5</SignType>
  <Sign>228c5368f3977bff06f7e9e4b687632c</Sign>
  <Version>1.0.0.0</Version>
  <Param>
    <OrderID>150714155309</OrderID>
    <Remark>处理成暂不能的备注</Remark>
  </Param>
</Request>
```

生成的待签名的字符串:

```
Action=ProblemhasbeenSolve&Param=Remark=处理成暂不能的备注&OrderID=150714155309
&Partner=jinriguoji1&RequestFmt=XML&ResponseFmt=XML&Version=1.0.0.0
签名结果=MD5( 待签名字符串+Key );
```

1.3.2.返回参数

Url 解码后的返回结果:

返回失败示例:

```
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageModel>
    <MessageCode>CallFail</MessageCode>
    <Description>操作失败的原因</Description>
  </MessageModel>
  <OrderID>150714155309</OrderID>
  <Sign>78b5d363ac8ba002f547bbc00d30733d</Sign>
</Response>
```

返回成功示例:

```
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageModel>
    <MessageCode>CallSuccess</MessageCode>
    <Description>操作成功</Description>
  </MessageModel>
  <OrderID>150714155309</OrderID>
  <Sign>78b5d363ac8ba002f547bbc00d30733d</Sign>
</Response>
```

三.附录说明

3.1 生成排序参数示例:

```
private static string LoopXmlNodeNodes2UrlParams(XmlNodeList nodes)
{
    List<string> kvs = new List<string>();
    var orderNodes = new List<XmlElement>();
    foreach (XmlElement node in nodes)
        orderNodes.Add(node);
    foreach (XmlElement node in orderNodes.OrderBy(o => o.Name))
    {
        if (node.IsEmpty) continue;
        if (!node.HasChildNodes)
            if (string.IsNullOrEmpty(node.InnerText))
                continue;
        string kv = string.Format("{0}=", node.Name);
        if (node.FirstChild is XmlText)
            kv += node.InnerText;
        else
        {
            var childKvs = LoopXmlNodeNodes2UrlParams(node.ChildNodes);
            if (childKvs == null) continue;
            kv += childKvs;
        }
        kvs.Add(kv);
    }
    return string.Join("&", kvs.OrderBy(o => o));
}
```

3.2 MD5 加密算法(C#示例)

```
/// <summary>
/// 获取32位长度的Md5摘要
/// </summary>
/// <param name="input"></param>
/// <param name="encoding"></param>
/// <returns></returns>
public static string Get32Md5(string input, Encoding encoding=null)
{
    if (encoding == null) encoding = Encoding.UTF8;
    StringBuilder buff = new StringBuilder(32);
    MD5 md5 = new MD5CryptoServiceProvider();
    byte[] t = md5.ComputeHash(encoding.GetBytes(input));
    foreach (byte t1 in t)
        buff.Append(t1.ToString("x").PadLeft(2, '0'));
    return buff.ToString();
}
```