

今通国际分销订单列表接口 (OpenApi)

文档版本信息

日期	文档版本	接口版本	描述	编撰者
2015-08-07	V1.0	V1.0.0.0	初稿	今通国际 研发中心

目录

文档版本信息.....	1
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.....	3
3.1.请求基本参数.....	3
3.2.返回基本参数.....	3
3.3.消息代码(MessageModel).....	3
3.4.标识代码.....	3
3.5.乘机人类型(PassengerType)	4
3.6.证件类型 (CardTypes).....	4
3.7.行李类型 (LuggageType)	4
3.8.订单状态 (OrderStates)	5
4. 执行过程.....	6
二.接口说明.....	7
1. 查询订单列表(GetOrderList).....	7
1.1.接口业务.....	7
1.2.参数说明.....	7
1.3.调用示例.....	9
1.3.1.请求参数.....	9
1.3.2.返回参数.....	9
2.附录.....	11
附 1: MD5 加密算法 (C#示例)	11
附 2: XML 报文生成签名串方法 (C#示例)	11

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。

测试环境地址: <http://testiapi.jinri.net:6012/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. 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

3.4. 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

3.5.乘机人类型(PassengerType)

值	描述
ADU	成人
CHD	儿童
INF	婴儿
ZZ	青年
SD	留学生 (STU)
DL	劳工 (LBR)
SL	移民 (EMI)
SC	海员 (SEA)
CD	老年
OT	探亲

3.6.证件类型 (CardTypes)

值	描述
IDCard	身份证
Passport	护照
MTP	台胞证
HKMOPermit	港澳通行证
HomeReturnCertificate	回乡证
SeamanCertificate	海员证
PMRTT	大陆居民往来台湾通行证

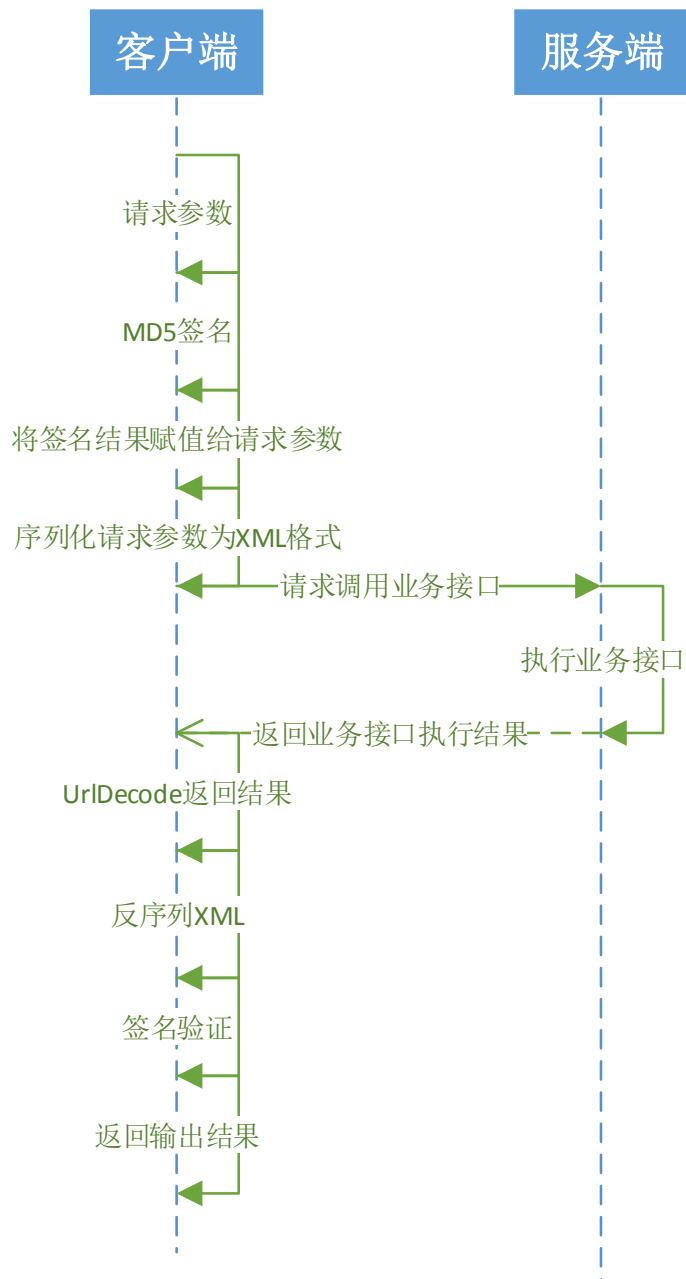
3.7.行李类型 (LuggageType)

值	描述
Outward	去程
SignCharge	回程
All	所有程

3.8. 订单状态 (OrderStates)

值	描述
A	平台审核
B	供应审核
C	等待支付
D	支付成功
E	出票完成
I	支付失败
J	审核退回
K	暂不能出票
L	已快钱支付到平台
M	退款处理中
N	退款完成
O	取消订单
P	取消订单
Q	暂不能废票
R	暂不能退票
S	取消成功-退款中
U	申请退票
V	申请废票
W	K 位政策
Z	改期升舱
G	未支付取消订单
-	空订单状态

4. 执行过程



二.接口说明

1. 查询订单列表(GetOrderList)

1.1.接口业务

该接口查询订单列表，含订单基本数据，航段信息，乘机人信息。

1.2.参数说明

接口名称		GetOrderList		
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	<u>OrderListRequestParam</u>	是	是	业务参数
业务参数: OrderListRequestParam				
字段名称	类型	是否必填		描述
Sdate	DateTime	是		查询开始时间(可精确到秒) 格式:yyyy-MM-ddTHH:mm:ss
Edate	DateTime	是		查询结束时间(可精确到秒) 格式:yyyy-MM-ddTHH:mm:ss
Consumer	string	是		分销商用户(此处传与 Partner 相同值)
Aircom	string	否		航空公司二字码
OrderID	string	否		订单号
PnrCode	string	否		PNR 编码
TicketNo	string	否		票号
FlightNo	string	否		航班号
PsgName	string	否		乘机人姓名
OrderStates	String	否		订单状态(多个状态以英文逗号分隔,如:E,D)
PageSize	int	是		单页数据量
PageIndex	int	是		页数
输出参数				
字段名称	类型	是否必填	是否签名	描述
MessageModel	<u>MessageModel</u>	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 <u>一.3.4</u>

Sign	string	是	否	返回参数签名结果
OrderCount	int	是	是	订单数量
PageSize	int	是	是	单页数据量(<=20)
TotalPage	int	是	是	总页数
CurrentPage	int	是	是	当前页
OrderList	List<OrderInfo>	是	是	订单信息集合
订单信息 OrderInfo				
字段名称	类型	是否必填	是否签名	描述
BaseOrder	BaseOrderInfo	是	是	订单基本信息
FlightList	List<FlightInfo>	是	是	航班行程信息
PassengerList	List<PassengerInfo>	是	是	乘机人信息
订单基本信息 BaseOrderInfo				
字段名称	类型	是否必填	是否签名	描述
OrderNo	String	是	是	订单号
ChangeComment	String	否	是	退改签规定
PnrCode	String	是	是	PNR 编码
TotalPrice	decimal	是	是	订单总价
OrderState	OrderStates	是	是	订单状态
PayTime	String	是	是	支付时间
PaymentWay	String	是	是	支付类型
Consumer	String	是	是	分销商
CreateTime	DateTime	是	是	订单创建时间
OrderComment	String	是	是	订单备注
PnrRtInfo	String	是	是	PNRRT 信息
航班信息 FlightInfo				
字段名称	类型	是否必填	是否签名	描述
Airline	String	是	是	航司（二字码）
FlightNo	String	是	是	航班号 MU9835
DepAirport	String	是	是	出发机场（三字码）
ArrAirport	String	是	是	抵达机场（三字码）
Cabin	String	是	是	舱位代码
DepTime	String	是	是	出发时间
ArrTime	String	是	是	到达时间
FlightType	String	是	是	航程往返类型(0: 默认;1: 去程;2: 回程)
乘机人信息 PassengerInfo				
字段名称	类型	是否必填	是否签名	描述
ID	Int	是	是	乘机人编号
Name	String	是	是	乘机人姓名
Sex	Int	是	是	性别(女=1,男=0)
PsgType	PassengerType	是	是	乘机人类型(参照 3.5 乘机人类型说明)
BirthDay	DateTime	是	是	生日

CardNo	String	是	是	证件号
CardType	CardTypes	是	是	证件类型
ValidDate	DateTime	是	是	证件有效期
TicketNo	String	是	是	票号
Nationality	String	是	是	国籍(二字码)
POI	String	是	是	证件签发地(国家二字码)

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>testrafree</Partner>
  <Action>GetOrderList</Action>
  <RequestFmt>XML</RequestFmt>
  <ResponseFmt>XML</ResponseFmt>
  <SignType>MD5</SignType>
  <Sign>04939244a53c33640c3eb004138e0d31</Sign>
  <Version>1.0.0.0</Version>
  <Param>
    <Consumer>testrafree</Consumer>
    <Sdate>2015-11-01T12:00:00</Sdate>
    <Edate>2015-11-16T12:00:00</Edate>
    <PageSize>10</PageSize>
    <PageIndex>1</PageIndex>
  </Param>
</Request>
```

生成的待签名的字符串:

```
Action=GetOrderList&Param=Consumer=testrafree&Edate=2015-11-16T12:00:00&PageIndex=1&Pa
geSize=10&Sdate=2015-11-01T12:00:00&Partner=testrafree&RequestFmt=XML&ResponseFmt=XM
L&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:

```
<?xml version="1.0"?>
```

```
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageModel>
    <MessageCode>CallSuccess</MessageCode>
    <Description>查询成功</Description>
  </MessageModel>
  <OrderCount>37</OrderCount>
  <PageSize>10</PageSize>
  <TotalPage>4</TotalPage>
  <CurrentPage>1</CurrentPage>
  <OrderList>
    <OrderInfo>
      <BaseOrder>
        <OrderNo>150805350712</OrderNo>
        <PnrCode>KSTKRX</PnrCode>
        <TotalPrice>1361.00</TotalPrice>
        <OrderState>A</OrderState>
        <PayTime>0001/1/1 0:00:00</PayTime>
        <Consumer>testrafree</Consumer>
        <CreateTime>2015-08-05T11:07:38.65</CreateTime>
      </BaseOrder>
      <FlightList>
        <FlightInfo>
          <Airline>MU</Airline>
          <FlightNo>MU742</FlightNo>
          <DepAirport>BKK</DepAirport>
          <ArrAirport>KMG</ArrAirport>
          <Cabin>R</Cabin>
          <DepTime>2015-08-28T15:45:00</DepTime>
          <ArrTime>2015-08-28T19:10:00</ArrTime>
          <FlightType>1</FlightType>
        </FlightInfo>
      </FlightList>
      <PassengerList>
        <PassengerInfo>
          <ID>206561</ID>
          <Name>CAO/YANG</Name>
          <Sex>0</Sex>
          <PsgType>ADU</PsgType>
          <BirthDay>1989-10-20T00:00:00</BirthDay>
          <CardNo>kJWOXSIH/9MYiaaY870olw==</CardNo>
          <CardType>IDCard</CardType>
          <ValidDate>2024-06-24T00:00:00</ValidDate>
          <TicketNo />
        </PassengerInfo>
      </PassengerList>
    </OrderInfo>
  </OrderList>
</Response>
```

```
<Nationality>CN</Nationality>
<POI> </POI>
</PassengerInfo>
</PassengerList>
</OrderInfo>
</OrderList>
</Response>
```

2.附录

附 1: 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();
}
```

附 2: XML 报文生成签名串方法 (C#示例)

```
/// <summary>
/// GetUrlParam
/// </summary>
/// <param name="xmlContent"></param>
/// <returns></returns>
public static string GetUrlParam(string xmlContent)
```

```
{
    XmlDocument doc = new XmlDocument();
    doc.LoadXml(xmlContent);
    var root = doc.DocumentElement;
    var urlParam = LoopXmlNodes2UrlParams(root.ChildNodes);
    return urlParam;
}

/// <summary>
/// XmlNodeList to urlParam
/// </summary>
/// <param name="nodes"></param>
/// <returns></returns>
private static string LoopXmlNodes2UrlParams(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.Name == "Sign" || node.Name == "SignType")
            continue;
        if (node.Name == "RequirePolicyCount")
            continue;

        if (!node.HasChildNodes)
            if (string.IsNullOrEmpty(node.InnerText))
                continue;
        string kv = string.Format("{0}=", node.Name);
        if (node.FirstChild is XmlText || node.FirstChild is XmlCDATASection)
            kv += node.InnerText;
        else
        {
            var childKvs = LoopXmlNodes2UrlParams(node.ChildNodes);
            if (childKvs == null) continue;
            kv += childKvs;
        }
        kvs.Add(kv);
    }
    return string.Join("&", kvs.OrderBy(o => o)); //注意是排序后生成连接串
}
```

