今通国际分销商票号推送接口

(OpenApi)

文档版本信息

日期	文档版本	接口版本	描述	编撰者
2015-07-20	V1.0	V1.0.0.0	初稿	今通国际研发中心
2015-12-15	V1.1	V1.0.0.0	票号信息增加证件号	今通国际研发中心
2016-04-11	V1.2	V1.0.0.0	增加 PnrCode	今通国际研发中心

目录

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.推送格式 http://ServerHostUrl/ticketNo.aspx?param=xml（该地址由分销商提供），保证接收参数为 param 即可 (注:param 参数值经 System.Web.HttpUtility.UrlEncode 处理,接收报文时需要 UrlDecode 处理)

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 不参与签名，其他参数包括业务参数均参与签名。

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）。

在接收到 XML 之后，需要按照签名规则进行签名比对，一致才可认为是有效的推送结果

3. 基本参数说明

3.1.xml 请求参数

参数	类型	是否为空	参数说明
OrderID	String	否	订单号
Info	List<TicketInfo>	否	票号信息
OutOrderNum	String	可空	外部订单号
PnrCode	String	可空	PNR 编码(存在换编时,有多个 PNR,第一个是旧编码,/后面是新编码)
Sign	String		签名
TicketInfo 信息			
参数	类型	是否为空	参数说明
PassengerName	String	否	乘机人姓名
CardNo	String	是	证件号
TicketCode	String	否	票号信息，多票号说明 如下
TicketCode 一人多票号说明: 217-2310064398-00 这个代表三个票号，拆开后为 217-2310064398, 217-2310064399, 217-2310064400 217-2310064398-99 这个代表两个票号，拆开后为 217-2310064398,217-2310064399			

对接票号通知接口时注意事项:

1. 乘机人的姓名，客户端在对接解析报文时，需要兼容处理乘机人标识,比如:MR/MS 后缀
2. 如果订单存在换编出票，推送的 PnrCode 会有两个 PNR,以/分隔，示例:75WM76/75WM76
3. XML 请求报文有作 HttpUtility.UrlEncode 编码处理，在收到报文后请作 HttpUtility.UrlDecode 解码处理。

3.2.编码前请求报文示例:

一个乘机人:

```
<?xml version="1.0"?>
<TicketInfoSOA>
   <OrderID>18041150984</OrderID>
   <Info>
      <TicketInfo>
         <PassengerName>XUE/BINGMS</PassengerName>
         <CardNo>EA1946192</CardNo>
         <TicketCode>160-2081659043</TicketCode>
      </TicketInfo>
   </Info>
   <OutOrderNum>18041109300066742227</OutOrderNum>
   <PnrCode>HYVEGQ</PnrCode>
</TicketInfoSOA>
```

待签名字符串(不含 KEY):

Info=TicketInfo=CardNo=EA1946192&PassengerName=XUE/BINGMS&TicketCode=160-2081659043&OrderID=18041150984&OutOrderNum=18041109300066742227&PnrCode=HYVEGQ

如果 KEY 为 1234567890123456

则需要进行 MD5 加密的字条串为:

Info=TicketInfo=CardNo=EA1946192&PassengerName=XUE/BINGMS&TicketCode=160-2081659043&OrderID=18041150984&OutOrderNum=18041109300066742227&PnrCode=HYVEGQ1234567890123456

多个乘机人:

```
<?xml version="1.0"?>
<TicketInfoSOA>
   <OrderID>18041150021</OrderID>
   <Info>
      <TicketInfo>
         <PassengerName>HUANG/ZHIPENG MR</PassengerName>
         <CardNo>EC4431919</CardNo>
         <TicketCode>001-5691737913</TicketCode>
      </TicketInfo>
      <TicketInfo>
         <PassengerName>YAO/BINXIA MS</PassengerName>
         <CardNo>E22461845</CardNo>
         <TicketCode>001-5691737914</TicketCode>
      </TicketInfo>
   </Info>
   <OutOrderNum>1289563251</OutOrderNum>
   <PnrCode>KSHSXV/LSXPKC</PnrCode>
</TicketInfoSOA>
```

待签名字符串(不含 KEY):

Info=TicketInfo=CardNo=E22461845&PassengerName=YAO/BINXIA MS&TicketCode=001-5691737914&TicketInfo=CardNo=EC4431919&PassengerName=HUANG/ZHIPENG MR&TicketCode=001-5691737913&OrderID=18041150021&OutOrderNum=1289563251&PnrCode=KSHSXV/LSXPKC

如果 KEY 为 1234567890123456

则需要进行 MD5 加密的字条串为:

Info=TicketInfo=CardNo=E22461845&PassengerName=YAO/BINXIA MS&TicketCode=001-5691737914&TicketInfo=CardNo=EC4431919&PassengerName=HUANG/ZHIPENG MR&TicketCode=001-5691737913&OrderID=18041150021&OutOrderNum=1289563251&PnrCode=KSHSXV/LSXPKC1234567890123456

3.3.返回参数及约定

参数	参数说明
SUCCESS	成功
FAIL	失败

成功请返回 SUCCESS(全大写), 如果未返回约定字符串，会认定为失败，会重复再次推送。

4. 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
 }
```