

今通国际分销商订单状态信息推送接口 (OpenApi)

文档版本信息

日期	文档版本	接口版本	描述	编撰者
2015-8-25	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	有变价时，推送信息增加改价原因	今通国际研发中心
2016-01-25	V1.2	V1.0.0.0	增加推送支付成功状态(D),但只通知供应商	今通国际研发中心
2016-04-11	V1.3	V1.0.0.0	增加 PnrCode	今通国际研发中心
2016-09-20	V1.4	V1.0.0.0	增加 AgioMoney, 今币通支付优惠金额(单人)	今通国际研发中心
2017-07-26	V1.5	V1.0.0.0	增加 Q/R 订单状态 暂不能信息推送	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. 适用场景说明.....	2
4. 基本参数说明.....	3
4.1.请求参数.....	3
4.2. 编码前请求报文示例.....	4
4.3.返回参数及约定.....	5
5. 附录.....	5
附 1: MD5 加密算法 (C#示例)	5
附 2: XML 报文生成签名串方法 (C#示例)	5

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/OrderStatusPush.aspx?param=xml` (该地址由分销商提供), 保证接收参数为 `param` 即可 (param 内容将作 `HttpUtility.UrlEncode` 编码处理)

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`).

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. 适用场景说明

分销商,针对 以下几种订单状态变化作推送通知, 等待支付(C)和审核退回(J), 暂不能废票(Q),暂不能退票(R)

4. 基本参数说明

4.1. 请求参数

参数	类型	是否为空	参数说明
OrderID	String	否	订单号
OrderState	String	否	订单状态(C=等待支付, J=审核退回, Q=暂不能废票, R=暂不能退票)
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)
PassengerInfo	List<PushTicketInfo>	可空	乘机人信息(含票号)
OrderPrice	List<Price>	可空	价格信息
OutOrderNum	String	可空	外部订单号
TotalCost	String	否	订单结算价,(订单状态为 J/Q/R 时, 此属性可空)
PlatMoney	decimal	否	平台管理费(单人)
AgiMoney	decimal	否	今币通支付优惠金额(单人)
ChangePriceReason	String	是	改价原因
ExtInfo	String	可空	状态为 J 时, 此处为审核退回原因, 为 Q 或 R 时 为暂不能原因
Sign	String	否	签名
PushTicketInfo 信息			
参数	类型	是否为空	参数说明
PassengerName	String	否	乘机人名称
CardNo	String	是	证件号
TicketCode	String	否	票号
PushPriceInfo 信息			
参数	类型	是否为空	参数说明
PassengerType	int	否	乘客类型 (0=ADU 成人, 1=CHD 儿童, 2=INF 婴儿, 3=ZZ 青年, 4=SD (STU) 留学生, 5=DL (LBR) 劳工, 6=SL (EMI) 移民, 7=SC (SEA) 海员, 8=CD 老年, 9=OT 探亲)
ExchangeRate	String	否	汇率

CurrencyCode	String	否	货币类型
FlightCost	String	否	票面价
TaxCost	String	否	税金
AgentRate	String	否	代理费率
AgioRate	String	否	返点奖励
AddMoney	String	否	开票手续费(单人)
AgioMoney	String	否	今币通支付优惠金额(单人)
FlightTotal	String	否	单张结算价
AdditionFlightCost	String	否	加段费
AdditionAgent	String	否	代理费

注意事项： xml 请求报文将作 `HttpUtility.UrlEncode` 编码处理.

在收到报文后请作 `HttpUtility.UrlDecode` 解码

4.2. 编码前请求报文示例

```

<PushOrderInfoSOA>
  <OutOrderNum>12358854</OutOrderNum>
  <OrderID>150825441452</OrderID>
  <OrderState>C</OrderState>
<PassengerInfo />
  <OrderPrice>
    <Price>
      <PassengerType>0</PassengerType>
      <ExchangeRate>1</ExchangeRate>
      <CurrencyCode>CNY</CurrencyCode>
      <FlightCost>27</FlightCost>
      <TaxCost>10</TaxCost>
      <AgentRate>3</AgentRate>
      <AgioRate>15</AgioRate>
      <AddMoney>2</AddMoney>
      <AgioMoney>0</AgioMoney>
      <AdditionFlightCost>6</AdditionFlightCost>
      <AdditionAgent>5</AdditionAgent>
    </Price>
  </OrderPrice>
  <TotalCost>35.00</TotalCost>
  <PlatMoney>2</PlatMoney>
<Sign>d3f5601c425ce20f58756efaa19b2894</Sign>
</PushOrderInfoSOA>
  
```

4.3.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。

5. 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 = LoopXmlNodeNodes2UrlParams(root.ChildNodes);
    return urlParam;
}

/// <summary>
/// XmlNodeList to urlParam
/// </summary>
/// <param name="nodes"></param>
/// <returns></returns>
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.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 = LoopXmlNodeNodes2UrlParams(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  
}
```