今通国际分销 PNR 文本生单政策校验接口 (OrderOpenApi)

文档版本信息

日期	文档版本	接口版本	描述	编撰者
2016-11-21	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.标识代码.....3
3.4.消息代码(MessageModel).....4
3.5.乘机人类型(PassengerType).....4
4. 执行过程.....5
二.接口说明.....6
1. PNR 文本生单政策校验(CreateOrderCheckPolicy).....6
1.1.接口业务.....6
1.2.参数说明.....6
1.3.调用示例.....7
1.3.1.请求参数.....7
1.3.2.返回参数.....10
2.附录.....11
附 1: PnrText 转 16 进制 加密方法(C#示例).....11
附 2: MD5 加密算法(C#示例).....11
附 3: XML 报文生成签名串方法(C#示例).....12

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/BookingOpenApi.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

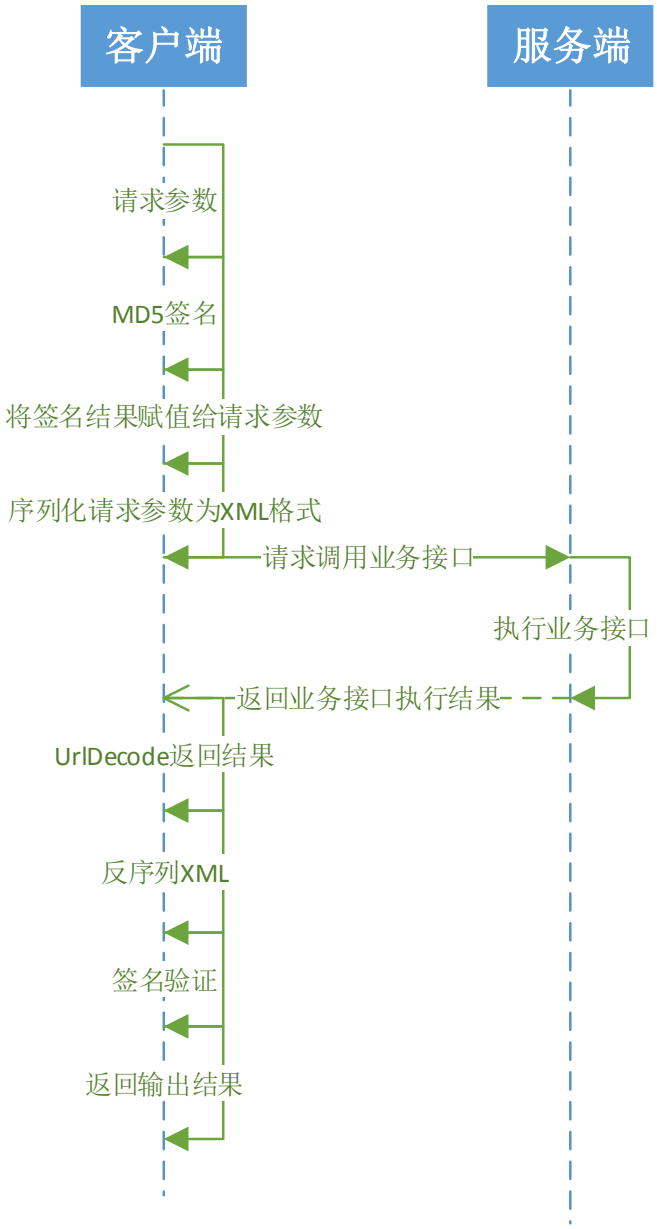
3.4.消息代码(MessageModel)

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
Desciption	string	对标识代码的描述内容

3.5.乘机人类型(PassengerType)

值	描述
ADU	成人, 0
CHD	儿童,1
INF	婴儿,2
CD	老年,3
SD	学生,4
DL	劳工,5
SL	移民,6
SC	海员,7
ZZ	青年,8
OT	其它,9

4. 执行过程


客户端
服务端
请求参数
MD5签名
将签名结果赋值给请求参数
序列化请求参数为XML格式
请求调用业务接口
执行业务接口
返回业务接口执行结果
UrlDecode返回结果
反序列XML
签名验证
返回输出结果

二.接口说明

1. PNR 文本生单政策校验(CreateOrderCheckPolicy)

1.1.接口业务

该接口根据 PNR 文本生单政策校验。

1.2.参数说明

接口名称		CreateOrderCheckPolicy		
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	CreateOrderRequestParam	是	是	业务参数
业务参数: CreateOrderRequestParam				
字段名称	类型	是否必填	描述	
Consumer	string	是	分销商用户名	
RouteInfo	CreateOrderModel	否	行程信息	
OrderText	CreatOrderTextParam	是	PNR 文本创单相关实体	
PnrText	string	是	PNR 文本(RT 文本) 加密成 16 进制数据，方法见附件说明	
Type	int	是	创单方式：0=RT 文本创单；1=行程信息创单(暂未开通)	
Out_OrderID	string	否	外部订单号	
SeatOffice	String	否	订座 Office 号(换编需要属性)	
SeatCTCT	String	否	订座 CT 电话(换编需要属性)	
业务参数: CreatOrderTextParam				
字段名称	类型	是否必填	描述	
PolicyID	string	是	政策 ID	
Linker	String	是	联系人	
LinkTel	string	是	联系电话	
Source	String	是	来源：0 默认 pnr 导入	
RepeatReason	string	否	重复导入原因	
PromptRemark	String	否	特殊提示	
AlterQtePrice	String	否	文件运价代码或路径	

SetNewPhone	String	是	电话号码
OrderComment	String	否	备注
SpePolicyParam	String	是	特价政策(包括境外价)生单参数, 对应政策返回数据
PriceFrom	String	否	运价来源
Passengers	List<Passenger>	是	乘机人信息
业务参数: Passenger			
字段名称	类型	是否必填	描述
Name	String	是	乘机人姓名,注:姓名将以 RT 文本中乘机人姓名为准，但此处的姓名最好与 RT 文本中一致(含乘机人相关标识),如: HU/LINA MS(订单入库 MS 标识会一起入库到乘机人表)如果此处与 RT 文本不一致,还会影响到票号通知接口。
Sex	String	是	乘机人性别 0=男,1=女
Type	String	是	乘机人类型(0=成人,1=儿童,2=婴儿)参照上面类型说明表
CardNo	String	是	证件号码，一般为护照
Valid	DateTime	是	证件有效期，校验证件有效期
Birthday	DateTime	是	生日， 校验生日有效期
Country	String	是	国籍 (默认：CN=中国)
Rate	int	是	机票打折率，不详请填 100
IsAddFlyer	int	是	常旅客（默认值：1）
输出参数			
字段名称	类型	是否必填	描述
MessageModel	MessageModel	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
PaySataus	String	是	订单状态,A=平台审核，C=自动审核
TicketPrice	String	否	成人票价(单人)
TotalTax	String	否	成人税(单人)
Sign	string	是	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:

```
<?xml version="1.0"?>
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Partner>测试帐号</Partner>  //注意：此处是对应开通的接口帐号
  <Action>CreateOrderCheckPolicy</Action>
```

```
<RequestFmt>XML</RequestFmt>
<ResponseFmt>XML</ResponseFmt>
<SignType>MD5</SignType>
<Sign>42788f361c2866a391b7c76e073a7dfd</Sign>
<Version>1.0.0.0</Version>
<Param>
  <Consumer>测试帐号</Consumer>  //注意：此处是对应开通的接口帐号
  <OrderText>
    <PolicyID>691CwSIQArI=</PolicyID>
    <Linker>张三</Linker>
    <LinkTel>021123456</LinkTel>
    <Source>0</Source>
    <RepeatReason />
    <PromptRemark />
    <AlterQtePrice />
    <SetNewPhone />
    <OrderComment />
    <PriceFrom />
<SpePolicyParam>F0USPoFb/rNUt9D0Uj2qyfWVdsttfeVN40LUZ3IDPrnGoan9VN0REQttG3RXvOJq/ZUkjRL9A8Mm
6c8CK3YjV+AeBHNdrkvQFqiUDmdH1y670sOi3bKWOMsskgCiRvi5LWO4J1KU5cMjcT3GCRc2ZBflcqkrJAGgqb8LKBJL
P8RGi9XX2awerZw05QqamoTeL6L8xta8Q/s=</SpePolicyParam>  //注意，此处为政策查询接口返回的对应值
    <Passengers>
      <Passenger>
        <Name>CAI/HUAHE</Name>
        <Sex>0</Sex>
        <Type>0</Type>
        <CardNo>HZ123</CardNo>
        <Valid>2018-12-31T00:00:00</Valid>
        <Birthday>1990-01-01T00:00:00</Birthday>
        <Country>CN</Country>
        <Rate>100</Rate>
        <IsAddFlyer>0</IsAddFlyer>
      </Passenger>
      <Passenger>
        <Name>WANG/YAN</Name>
        <Sex>0</Sex>
        <Type>0</Type>
        <CardNo>HZ456</CardNo>
        <Valid>2017-12-31T00:00:00</Valid>
        <Birthday>1980-05-01T00:00:00</Birthday>
        <Country>CN</Country>
        <Rate>100</Rate>
        <IsAddFlyer>0</IsAddFlyer>
      </Passenger>
```

```
        </Passengers>
      </OrderText>

<PnrText>52544850515250512020202020202020202020202020202020202020202020202020202020202020202
02020202020202020202020202020202020202020202020202020202020202020202020202020DA20312E4341492F4
85541484520322E57414E472F59414E204850515250512020202020202020202020202020202020202020202020202
020202020202020202020202020202020202020202020202020202020DA20332E2020435A33303335204C20202053553
1394A554C202043414E424B4B20484B3220202031353035203136353020202020202020202045202D2D3120202
02020202020202020202020202020202020DA20342E2020435A33353820204C2020205341303141554720204B4B434
14E20484B32202020313135352031353435202020202020202020204520312020D2D20202020202020202020202020202
020202020DA20352E5A55482F54205A55482F5420303735362D383132333138302F5A55482051494E47205955414
E20414952205345525649434520434F2E2F4C4955594920414243444546472020202020202020DA20362E544C2
F313330352F31394A554C2F5A5548313033202020202020202020202020202020202020202020202020202020202
020202020202020202020202020202020202020202020202020202020202020DA20372E535352204144544B20314520425
9205A554831334A554C31352F32333032204F522043584C20435A20424F4F4B494E47202020202020202020202
0202020202020202020202020202020202020DA20382E53535220444F435320435A20484B3120502F434E2F4533353
439323432352F434E2F313053455038392F462F32354A414E32342F57414E472F59414E2F503220202020202020202
02020202020DA20392E53535220444F435320435A20484B3120502F434E2F45353130373333939382F434E2F313
44A414E39312F462F31324D415932352F4341492F48554148452F50312020202020202020202020202020DA31302E4
F534920435A2043435443543138393235353830353539202020202020202020202020202020202020202020202020202
020202020202020202020202020202020202020202020202020202020202020DA31312E4F534920435A20435443543
1383932353538303535392020202020202020202020202020202020202020202020202020202020202020202020202
020202020202020202020202020202020202020DA31322E524D4B2043412F4D5842524D582020202020202020202
0202020202020202020202020202020202020202020202020202020202020202020202020202020202020202020202
020202020202020DA31332E5A5548313033</PnrText>
        <Type>0</Type>
        <Out_OrderID>201608926</Out_OrderID> //注意，外部订单号 需要有值 才能有相关订单通知
        <SeatOffice/>
        <SeatCTCT/>
    </Param>
</Request>
```

生成的待签名的字符串:

Action=CreateOrderCheckPolicy&Param=Consumer=测试环境帐号&OrderText=Linker=张三
&LinkTel=021123456&Passengers=Passenger=Birthday=1980-05-01T00:00:00&CardNo=HZ45
6&Country=CN&IsAddFlyer=0&Name=WANG/YAN&Rate=100&Sex=0&Type=0&Valid=2017-12-31T0
0:00:00&Passenger=Birthday=1990-01-01T00:00:00&CardNo=HZ123&Country=CN&IsAddFlye
r=0&Name=CAI/HUAHE&Rate=100&Sex=0&Type=0&Valid=2018-12-31T00:00:00&PolicyID=69l
CwSIQArI=&Source=0&SpePolicyParam=F0USPoFb/rNUt9D0Uj2qyfWVdsttfeVN40LUZ3IDPrnG
oan9VN0REQttG3RXvOJq/ZUkjRL9A8Mm6c8CK3YjV+AeBHNdrkvQFqiUDmdH1y670sOi3bKWOMs
skgCiRvi5LWO4J1KU5cMjcT3GCRc2ZBflcqkrJAGgqb8LKBJLP8RGi9XX2awerZw05QqamoTeL6
L8xta8Q/s=&Out_OrderID=201608926&PnrText=52544850515250512020202020202020
20202020202020202020202020202020202020202020202020202020202020202020202020202

0202020202020202020202020202020202020202020DA20312E4341492F485541484520
322E57414E472F59414E204850515250512020202020202020202020202020202020202020202
0202020202020202020202020202020202020202020202020202020202020DA20332E2020435A
33303335204C202020535531394A554C202043414E424B4B20484B32202020313530352031 3
6353020202020202020202020202045202D2D312020202020202020202020202020202020202020DA20
342E2020435A33353820204C20202053413031415547202042 4B4B43414E20484B322020203
1313535203135343520202020202020202020202045203120 2D2D202020202020202020202020202020
20202020DA20352E5A55482F54205A55482F5420303735362D383132333138302F5A5548205
1494E47205955414E204149522053455256494345204 34F2E2F4C4955594920414243444546
472020202020202020202020DA20362E544C2F313330352F31394A554C2F5A55483130332020202
0202020202020202020202020202020202020202020202020202020202020202020202020202020
2020202020202020202020202020202020DA20372E535352204144544B203145204259205A554
831334A554C31352F32333032204F522043584C20435A20424F4F4B494E4720202020202020
2020202020202020202020202020202020202020202020DA20382E53535220444F435320435A2
0484B3120502F434E2F45333534393234323 52F434E2F313053455038392F462F32354A414E
32342F57414E472F59414E2F503220202020202020202020202020202020DA20392E53535220444
F435320435A20484B3120502F434E2F45353130373333939382F434E2F31344A414E39312F46
2F31324D415932352F4341492F48554148452F503120202020202020202020202020202020DA31302
E4F534920435A204343544354313839323535383035353920202020202020202020202020202020
2020202020202020202020202020202020202020202020202020202020202020202020202020202
02020DA31312E4F534920435A204354435431383932353538303535392020202020202020202020
2020202020202020202020202020202020202020202020202020202020202020202020202020202
0202020202020202020202020202020202020202020202020202020202020DA31322E524D4B2043412F4D5842524D582020202020202020202020202020
2020202020202020202020202020202020202020202020202020202020202020202020202020202
0202020202020202020202020202020DA31332E5A5548313033 3&Type=0&Partner=测试环境帐号
&RequestFmt=XML&ResponseFmt=XML&Version=1.0.0.0

注意: Passengers 节点里的子节点也要排序，生成连接串方法参照下面的 LoopXmlNodes2UrlParams

签名结果=MD5(待签名字符串+Key);
示例: 如果待签名字符串为 abcdef, Key 为 123456,
签名结果 MD5(abcdef123456)为: 9ee721fdedf4175983cae42b2b741b89

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> //CallSuccess成功, 更多详见3.3标识代码说明
    <Desciption>校验成功</Desciption>
```

```
  </MessageModel>
  <PaySataus>C</PaySataus> //订单状态, A 是平台审核, C 是自动审核
   <TicketPrice>-1</TicketPrice> //成人票价(单人)
  <TotalTax>-1</TotalTax>  //成人税(单人)
  <sw_time>0</sw_time>
</Response>
```

2.附录

附 1: PnrText 转 16 进制 加密方法(C#示例)

注意此加密算法，在有些环境下编码不一致可能会导致解码后会出现乱码现象
验证加解密是否一致测试小工具 http://testiapi.jinri.net:6012/tool/textConvert.aspx

```
/// <summary>
/// 加密X2
/// </summary>
/// <param name="str"></param>
/// <returns></returns>
public static string StrToHex(string str)
{
 string strTemp = "";
 if (string.IsNullOrWhiteSpace(str))
      return "";
 byte[] bTemp = Encoding.Default.GetBytes(str);

for (int i = 0; i < bTemp.Length; i++)
{
   //注意，也可以写成bTemp[i].ToString("X2");
   //用以返回两位的16进制数，X有大小写之分
    strTemp += bTemp[i].ToString("X");
 }
   return strTemp;
}
```

附 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();
}
```

附 3: 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
}
```