

今通国际分销 PNR 文本创订单接口 (OrderOpenApi)

文档版本信息

日期	文档版本	接口版本	描述	编撰者
2015-8-3	V1.0	V1.0.0.0	初稿	今通国际研发中心
2016-03-29	V1.1	V1.0.0.0	增加换编需要属性 SeatOffice 及 SeatCTCT	今通国际研发中心
2016-04-19	V1.2	V1.0.0.0	增加校验证件有效期及生日有效期	今通国际研发中心
2016-08-16	V1.3	V1.0.0.0	增加特价政策(包括境外价)生单参数	今通国际研发中心
2016-09-27	V1.4	V1.0.0.0	优化了境外政策的境外运价	今通国际研发中心
2017-12-28	V1.5	V1.0.0.0	增加 Ed 节点	今通国际研发中心
2018-07-18	V1.6	V1.0.0.0	增加返回 AllowPayTypes	今通国际研发中心
2018-11-28	V1.7	V1.0.0.0	增加乘机相关参数, Tel, CardType, CardIssuePlace	今通国际研发中心

目录

文档版本信息.....	1
目录.....	2
一.调用过程.....	3
1. 请求方式.....	3
2. 签名方式.....	3
3. 基本参数说明.....	4
3.1.请求基本参数.....	4
3.2.返回基本参数.....	4
3.3.标识代码.....	4
3.4.消息代码(MessageModel).....	5
3.5.乘机人类型(PassengerType).....	5
4. 执行过程.....	6
二.接口说明.....	7
1. PNR 文本创单(CreateOrderByPnrText).....	7
1.1.接口业务.....	7
1.2.参数说明.....	7
1.3.调用示例.....	9
2.附录.....	13
附 1: PnrText 转 16 进制 加密方法(C#示例).....	13
附 2: MD5 加密算法 (C#示例)	14
附 3: XML 报文生成签名串方法 (C#示例)	14

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	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	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
CallSuccess	成功
Unknown	未知错误
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
Unauthorized	无权限的访问
AnalyzePnrFail	PNR 解析失败
MissFlightType	解析 PNR 格式有误(缺少航班类型)
SegmentIsNull	航段不能为空
DepCodeError	出发机场三字码不正确

ArrCodeError	到达机场三字码不正确
FlightDateError	航班日期不正确
FlightTypeError	航班类型不正确
NonePolicy	没有符合条件的政策
PassengerIsNull	乘机人不能为空
PassengerNameError	乘机人姓名格式不对(包括不能为空)
PassengerTypeError	乘机人类型不对
PassengerSexError	乘机人性别不对(包括不能为空)
PassengerCardTypeError	乘机人证件类型不能为空(包括数据校验)
PassengerCardIsNull	乘机人证件号不能为空
PassengerCardError	乘机人证件号格式不对
PassengerBirthdayIsNull	乘机人生日不能为空
PassengerCardValidError	乘机人证件有效期不能为空
PassengerBirthdayError	乘机人类型与生日校验不一致
NoSeat	您所选价格已无位
PassengerRepeatExist	乘机人重复订位生单
ProviderOrderError	生单失败(供应侧生单失败)
PolicyVerifyFail	政策校验失败

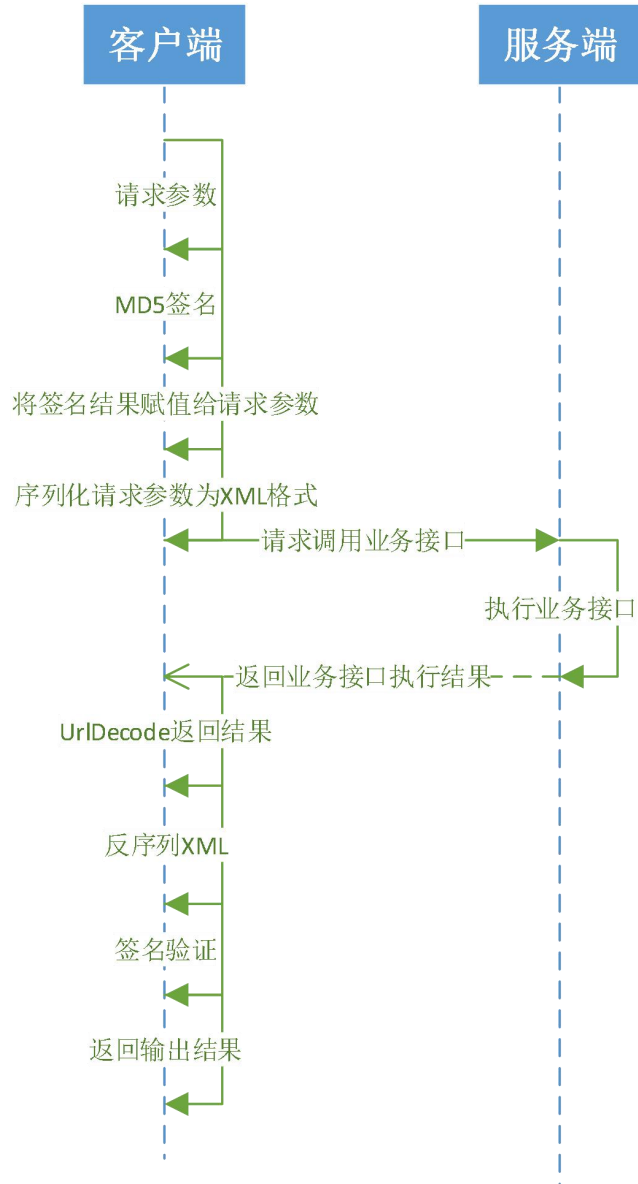
3.4.消息代码(MessageModel)

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

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



二.接口说明

1. PNR 文本创单(CreateOrderByPnrText)

1.1.接口业务

该接口根据 PNR 文本创建订单。

1.2.参数说明

请求参数

接口名称		CreateOrderByPnrText		
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, RT 文本里航班号数字部份请至少补全 2 位, 例如:航班 EY1, 传入示例:EY01, 可支持格式:EY01 或 EY001 或 EY0001	
Type	int	是	创单方式: 0=RT 文本创单; 1=行程信息创单	
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	
Ed	string	是	政策接口返回的政策列表里的 ed 内容	
Linker	String	是	联系人	
LinkTel	string	是	联系电话	
Source	int	是	订单来源(0= 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	否	联系手机号
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
Passenger Element			
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, 3=青年, 4=留学生, 5=劳工, 6=移民, 7=海员, 8=老年, 9=探亲)
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 格式: 2019-12-31T00:00:00
Birthday	DateTime	是	生日, 校验生日有效期 格式: 1990-02-08T00:00:00
Country	String	是	国籍 (默认: CN=中国)
Rate	int	是	机票打折率, 不详请填写 100
IsAddFlyer	int	是	常旅客 (默认值: 1)
Tel	string	否	乘机人电话
CardType	string	否	证件类型: PP=护照, GA=港澳通行证, TW=台湾通行证, TB=台胞证, HX=回乡证, HY=国际海员证, 不填将默认 PP(护照)
CardIssuePlace	string	否	证件发行国家, 国家二字码, 此项不填将默认与国籍 Country 一致

返回参数

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 二.3.4
OrderNo	String	是	订单号
PayPrice	String	否	订单总金额, 订单状态为 C 时, 此项有值
PaySataus	String	是	订单状态,订单状态为 C 时可以直接支付,其它状态需要等待订单状态通知接口通知 C 状态后方可支付

AllowPayTypes	String	是	支持的支付方式(多个以[,]英文逗号分隔) 示例: AliPay, JinPay
PNR	String	否	返回 PNR, 可能为空, 可接入订单详情接口获取
TicketPrice	String	否	暂无, 可接入订单详情接口获取
TotalTax	String	否	暂无, 可接入订单详情接口获取
Sign	string	是	返回参数签名结果

注意:

接口重复订单处理逻辑: 在一定时间内(默认是 15 分钟), 相同的乘机人信息及政策 ID 生单, 如果未调用取消订单接口。接口会返回第一次请求时生成的订单号, 乘机人信息包括: 乘机人姓名, 行程信息, PNR。

调用了取消订单接口后, 接口会重新生成订单。

1.3.调用示例

1.3.1.请求参数

示例报文:

```
<?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>CreateOrderByPnrText</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> //注意: 此处是对应政策查询接口返回的政策ID
      <Ed>FF93DCF7CBA14D9D05DF02E88603AC4F66D8849FBB325D9D13A5228FA0C1AADBE4AFFE7F557B263EAC
B0BF63F32E02C03D4A0D1B4C5FBA42B53C2D136C7937B811EFB475336BDF58B9F480B93B04E9075E52665237FB93
9147FEC261A4F0D4223D90CF1E0BD02564DD821B32E9A4A1F93492C58BF285847D58C10C0C12CDD1BE3D33390F51
90E613BF4A77DA914F9C1FA28BF89259C62EF7BBFF2B67D245AF295F88C244770902780C3B351584D34000CBA4FB
EECC0BBB357280A6635514B5FA76D1B0043EC46622563BF35562DCB2EB68C51721C2B092C27B9F411A50B0778DC1
34FE93333F9D0792F192B6B5B3718E1E2C843BB6F02F4CCA69F36B587F24F5DDE517A61FB6EFB773199543EB5FAF
5D0D8B3FAAC8662696C7025201C10C5F9CBD953E00CA4A6380C5929574508B24052C74E10D8F758F11835E97E190
CF02D74DD5EFA4BDA1712BFD232FCE0A57A105C601E6774C9EBE45009F9F5CB0DB99ED5B6AA9CF179ECC409D83D9
930BFF4B380906023A7CFFA63DEA4A0B3228222F53F7482488CA3CDB1BBBEA315557CC24E4A7241469DF1CF147D4
2B76C78BA3A4EEDF20379B7EC8082E30E86CFC1A39A26A09DC3EBB1954E615CCC566261320ADD2F07F8CD988ABB3
65825B7D5018BB92133A4D2DB36D16B17C74796F0CF59875AD084115C52DA751B79194F435258C0AF109A582520D
40278A8F0342E7ABA3D57457FB896D92DE84C5D0C326A6E34EC8AF953D4416761DEC8AECB25EFBC146A51F1E58C2
2F37FE548388F40E445C4008399024B8644E1BA82418136EF878E5249F3C03C05184A9DB5C2D013BBCE9C845669B
```

[illegible]

</Request>

[illegible]


```
<OrderNo>150812833484</OrderNo> //订单号
<PayPrice>-1</PayPrice> //订单总金额, 订单状态为C时, 此项有值
<PaySataus>C</PaySataus> //订单状态, 订单状态为C 时可以直接支付, 其它状态需要等待订单状态通
知接口通知C状态后方可支付
<PNR>TEST74</PNR> //返回PNR, 可能为空, 可接入订单详情接口获取
<TicketPrice>-1</TicketPrice> //暂无, 可接入订单详情接口获取
<TotalTax>-1</TotalTax> //暂无, 可接入订单详情接口获取
<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://testapi.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.IsNullOrEmpty(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 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
}
```