

## 今通国际分销 PNR 文本获取政策接口 (BookingOpenApi)

### 文档版本信息

日期	文档版本	接口版本	描述	编撰者
2015-08-03	V1.0	V1.0.0.0	初稿	今通国际研发中心
2015-11-26	V1.1	V1.0.0.0	修订, 增加获取政策数量参数	今通国际研发中心
2016-08-16	V1.2	V1.0.0.0	修订, 增加特价政策及境外价参数	今通国际研发中心
2016-09-27	V1.3	V1.0.0.0	优化了境外政策的境外运价	今通国际研发中心
2016-10-13	V1.4	V1.0.0.0	政策详情节点增加协议 PrivilegeValue 和 PlatFee	今通国际研发中心
2017-04-20	V1.5	V1.0.0.0	政策请求报文协议 LowerWaitTime	今通国际研发中心
2017-09-11	V1.6	V1.0.0.0	更新政策适用时间使用说明, <b>pbegtime</b> 和 <b>pendtime</b> 都为 00:00 时, 供应未设置上下班时间, 政策不可用, 两个时间都相同时, 政策也不可用	今通国际研发中心
2017-09-12	V1.6	V1.0.0.0	增加 Apply_Nationality 属性, 修改 TransitAirport 和 IsWaste 属性描述	今通国际研发中心
2018-01-12	V1.6	V1.0.0.0	增加 Pnr 行程信息(含行李额)	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.....	3
3.1.请求基本参数.....	3
3.2.返回基本参数.....	4
3.3.消息代码(MessageModel).....	4
3.4.标识代码.....	4
3.5.乘机人类型(PassengerType) .....	5
4. 执行过程.....	6
二.接口说明.....	7
1. PNR 文本获取政策(GetPolicyByPnrText).....	7
1.1.接口业务.....	7
1.2.参数说明.....	7
1.3.调用示例.....	12
2.附录.....	18
附 1: PnrText 转 16 进制 加密方法(C#示例) .....	18
附 2: MD5 加密算法 (C#示例) .....	18
附 3: XML 报文生成签名串方法 (C#示例) .....	19

## 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)。

### 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	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	<a href="#">MessageCode</a>	执行结果标识代码,详见 <a href="#">3.3.标识代码</a>
Description	string	对标识代码的描述内容

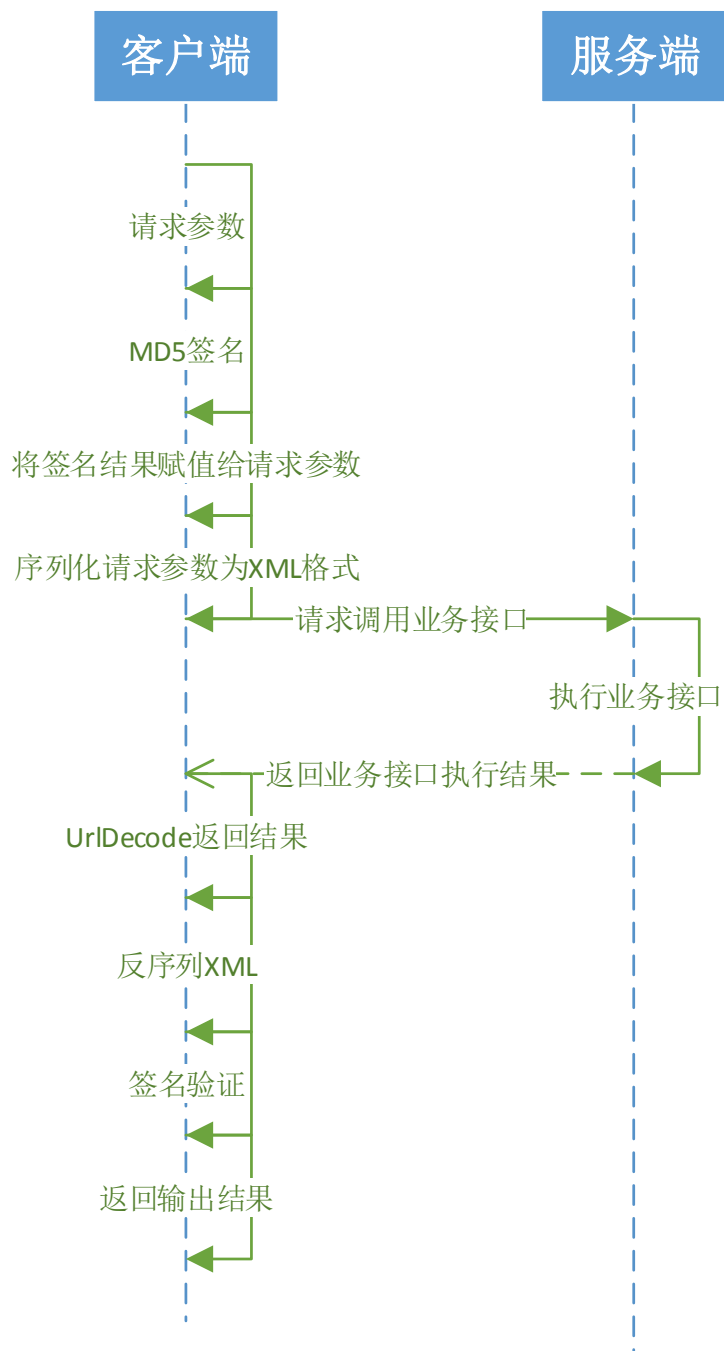
### 3.4.标识代码

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

### 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
DL	劳工
EMI	移民
SC	海员
CD	老年
OT	探亲

## 4. 执行过程



## 二.接口说明

### 1. PNR 文本获取政策(GetPolicyByPnrText)

#### 1.1.接口业务

该接口根据 PNR 文本获取政策数据。

#### 1.2.参数说明

请求参数

接口名称		GetPolicyByPnrText		
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	<a href="#">PNRTextGetPolicyRequestParam</a>	是	是	业务参数
业务参数:PNRTextGetPolicyRequestParam				
字段名称	类型	是否必填	描述	
UserName	string	是	用户名	
ChangePnr	bool	是	是否获取 需换编政策 true/false(注:目前特价政策都是要换编出票, 如果需要特价政策,此项 true)	
Category	bool	是	是否获取 境外电子政策 true/false	
NoRMB	bool	是	是否获取 非人民币币种政策 true/false	
IsGetSingle	bool	是	是否获取一条最优政策 true/false	
RequirePolicyCount	int	是	获取政策数量 默认 0(当为 0 时返回 1 条最优政策, 最大 3,代表每种类型政策返回的数据量, 目前为 3 时最多返回 9 条政策) 注:此参数不参与签名	
LowerWaitTime	int	否	获取特价政策等待时间(单位秒)默认 0	
PnrContent	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	

返回参数

字段名称	类型	是否必填	是否签名	描述
MessageModel	<a href="#">MessageModel</a>	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 <a href="#">一.3.4</a>
Price	<a href="#">PolicyPrice</a>	是	是	成人价格信息(公布运价) 参照下面 PolicyPrice Element
PriceList	<a href="#">List&lt;PolicyPrice&gt;</a>	否	是	儿童及婴儿价格信息(公布运价) 参照下面 PolicyPrice Element
PolicyList	<a href="#">List&lt;PolicyInfo&gt;</a>	是	是	政策信息 PolicyInfo Element
PnrFlightList	<a href="#">List&lt;PnrFlightInfo&gt;</a>	是	是	行程信息 PnrFlightInfo Element
Sign	string	是	否	返回参数签名结果
行程信息 PnrFlightInfo Element				
字段名称	类型	是否必填	描述	
AirCompany	string	是	航司二字码	
Start	string	是	出发机场（三字码）	
End	string	是	到达机场（三字码）	
StartCity	string	是	出发城市码	
EndCity	string	是	到达城市码	
FareBasic	string	是	运价基础代码	
FareBasicFSU	string	是	运价基础代码 FSU	
Nubmer	string	是	航班号	
FlightDate	string	是	起飞日期	
FlightFTime	string	是	起飞时间	
FlightTTime	string	是	到达时间	
Cab	string	是	舱位	
CabChild	string	是	子舱位标识	
FlightState	string	是	航班状态 HK1 NN 等	
ShareCompany	string	否	共享航司/空	
IsShare	bool	是	是否共享航班	
IsOpen	bool	是	是否 OPEN 行程	
IsMain	string	否	是否主航段 否 0 是 1	
Type	string	否	航班类型（单程 0 去程 0 返程 1 其他段 2）	
luggage	string	否	行李额 1PC/2PC/20K/30K	
价格信息 PolicyPrice Element				
字段名称	类型	是否必填	描述	
Type	int	是	运价类型，成人=0, 儿童=1, 婴儿=2	
Fare	decimal	是	运价	
Tax	decimal	是	税费	
Total	decimal	是	总价格	
IsQPoint	bool	是	是否有 Q 税	
Qpoint	decimal	是	Q 税金(单位:元)	
IT	bool	是	是否 IT 票面	
NetFare	bool	是	是否 NET 运价	
Commision	decimal	是	返点值	
政策信息 PolicyInfo Element				
字段名称	类型	是否必填	描述	



id	String	是	政策编号 不可空提交订单接口时需将此 ID 作为参数, 原值传回
ed	String	是	创单时使用, 原值传回
Airline	String	是	航空公司 二字码
DepCity	String	是	出发机场三字码
ArrCity	String	是	到达机场三字码
SaleSDate	String	是	政策有效期(开始时间)
SaleEDate	String	是	政策有效期(结束时间)
Discount	String	是	代理费 数字 不可空
Cabin	String	是	舱位
Agio	String	是	奖励 数字
AddMoney	String	是	开票费
CurrencyCode	String	是	币种编码
RateValue	String	是	汇率值
FlightType	String	是	政策类型 (0 单程, 1 返回)
TicketS	String	否	开票有效期(开始日期)
TicketE	String	是	开票有效期(结束日期)
Remark	String	是	政策备注
eReturntrip	String	是	回程结束时间(回程有效期延用至)
officeid	String	是	授权 office
isCHangePnr	string	否	是否需换编 D=可空无需换编码(D) A=换编码出票(A) B=大编出票(无需换编)(B) C=大编出票(需换编)(C) E=外放无位不换编/但外放有位时需换编
UpdateDate	string	否	政策更新时间戳
Z	string	是	是否中转政策(否 0 是 1)
ZDepCity	string	否	中转机场 Z 为 0 时, 为空
ZCabin	string	否	中转仓位 Z 为 0 时, 为空
IsCategory	string	否	票本类型 BSP(A) B2B(B) 境外电子(C) BOP(D)
IsWaste	string	否	是否允许作废 允许(空) 不允许(A) 执行废票及退改以政策备注(Remark)为准
IsChildren	string	否	儿童票 奖励与成人一致(A) 可开无奖励(B) 不可开(C) 指定奖励(D), 注意: 当行程中有乘客儿童, 政策中该字段为【可开无奖励(B)】时, 该政策奖励为 0, 为【不可开(C)】时该政策不适用, 为【指定奖励(D)】时, 儿童奖励值用 ChildrenRate 字段
IsBaby	String	否	婴儿票 可开无奖励(A) 不可开(B) 注意: 当行程中有乘客婴儿, 政策中为【可开无奖励(A)】时, 该政策奖励为 0, 为【不可开(B)】时, 该政策不适用

IsShare	String	否	共享航司 同时适用的航司, 多个以, 分割
IsAdapt	String	否	返点不适用 即: 乘客中包含以下类型时无奖励【留学生(A) 移民(B) 劳工(C) 海员(D) 青年(E) 老年(R) 探亲(F) 港澳台(T) 外籍(M) 非汉字姓名(U) Add-On(V) 缺口程(G) OPEN(N) SPA(P) Q值(S) 共享航班(O) 三方协议(H) 套票(I) 整团(J) 小团(K)】
Price	String	否	运价限制 票面(不含税)高于 Price 才有奖励
isreward	String	否	是否允许 1/2RT 混舱是(A) 否(空)
IsAuto	String	否	是否允许自动出票, 不允许 0, 允许 1
AdvanceDays	String	是	提前开票天数
pbegtime	String	是	开票开始时间, 示例 08:00
pendtime	String	是	开票结束时间, 示例 20:00
注: pbegtime 和 pendtime 都为 00:00 时, 供应未设置上下班时间, 政策不可用, 两个时间都相同时, 政策也不可			
pbegtime_	String	是	午休开始时间, 示例 12:00
pendtime_	String	是	午休结束时间, 示例 13:00
pbegtime_i	String	是	自动出票开始时间, 示例 00:01
pendtime_i	String	是	自动出票结束时间, 示例 23:59
OneSubtracting	String	否	单程直减 数字
RoundSubtracting	String	否	往返直减 数字
ChildrenSingle	String	否	儿童不单开:0 不选, 1 不单开, 值为 1 时, 如果行程中为单独乘客儿童, 该政策不适用
ChildrenRate	String	否	儿童指定奖励 请参考 IsChildren 说明
ChildrenFee	String	否	儿童加收手续费
BabyFee	String	否	婴儿加收手续费
Apply_Nationality	String	否	不适用国籍
Apply_AgeStart	String	否	适用年龄段 左值(0 代表不限)
Apply_AgeEnd	String	否	适用年龄段 右值(0 代表不限)
Apply_People	String	否	适用人数(*人以上) 0 代表不限, 如果有儿童乘客, 则看 IsChildren 字段是否为【奖励与成人一致(A)】, 如果是(A), 则可以将儿童算是成人名额, 否则只计算成人名额
OnlyPeopleEven	String	否	只适用双数否(不限)0 是 1, 如果有儿童乘客, 则看 IsChildren 字段是否为【奖励与成人一致(A)】, 如果是(A), 则可以将儿童算是成人名额, 否则只计算成人名额
ChangePnrAuthorize	String	否	换编不授权
NoOutTicketOffice	String	否	指定 Office 不开票
FareBasisCodeInc	String	否	运价基础中含的选项 无奖励 1 无代理费 2

lude_NOClose			
FareBasisCodeInclude	String	否	运价基础中含的值 例如:PR/WX/WR/, 与 FareBasisCodeInclude_NOClose 配合使用
ITPriceNoReward	String	否	IT 票面 不限 0 无奖励 1
ITPriceMoreAddFee	String	否	IT 票面 加收手续费
FareBasisCode	String	否	运价基础需一致 (不限 0, 需一致 1)
BelowPar	String	否	票面略低 否 0 是 1
NoOutTicketOfficeInfo	String	否	指定 office 不开票或换编出票 0 不开票 1 换编出票, (如果导入 PNR 的行程带了订位 office 号, 而该字段为【不开票 0】, 同时 NoOutTicketOffice 字段中包含该 Office 号, 则该政策不适用, 如果该字段为【换编出票 1】, 同时 NoOutTicketOffice 字段中包含该 Office 号, 则该政策需换编出票)
RoundMustSpace	String	否	往返须包含舱位, 如: F/E/Y/ (空表示不限, 否则国际主段的舱位必须存在【往返必须包含舱位】里面。不在, 则将奖励置为 0。)
MustProviderReview	String	否	需供应审核 否 0 是 1
NotApplicableAirLine	String	否	不适用航线
RetrunAirportType	String	否	返回机场范围类型 (0: 同一洲际范围内, 1: 政策始发机场范围, 2: 必须返回行程出发机场, 3: 指定返回国家或机场范围)
RetrunAirport	String	否	指定返回国家或机场范围 CN/JP/PVG/SHA/HKG/
TourCode	String	否	旅行代码 (完整或部分)
TransitAirport	String	否	不适用转机机场 (PVG/SHA/HKG/)
ExFlights	<a href="#">List&lt;ExFlight&gt;</a>	否	(不) 适用航班集注意: 当没有限制航班时, ExFlights 节点不存在
UnTimes	<a href="#">List&lt;UnTime&gt;</a>	否	不适用日期集当没有不适用日期时, UnTimes 节点不存在
PTs	<a href="#">List&lt;PType&gt;</a>	否	乘机人类型集, 此节点各乘机人类型对应的返点及奖励都可以直接使用。乘机人类型对应结算价可供参考, 均为人民币。(生单若有变价, 以生单返回的为准。)
Effective	String	是	是否有效, 即政策是否可用, 0=有效, 1=有效(需审核), 2=有效(儿童或婴儿无奖励), 3=无效, (非有效状态, 一般可在政策备注中查询原因, 在匹配处理逻辑时, 状态为 2 不用再对 Agio 作特殊处理, PType 里各乘机人类型对应的返点

			及奖励都可以直接使用)
SpePolicyParam	String	是	特价政策及境外政策必要属性 (生单时必要属性)
SpePolicyType	int	是	特价政策类型 (0=普通政策, 1=特价政策)
PrivilegeValue	decimal	是	今币通支付优惠金额
PlatFee	decimal	是	平台管理费
不适用航班 ExFlight Element			
字段名称	类型	是否必填	描述
Type	String	否	不适用航班 0, 去程适用航班 1, 回程适用航班 3
Flight	String	否	航班号, CA1234/CA3213/
不适用日期 UnTime Element			
字段名称	类型	是否必填	描述
Type	String	是	类型 (0 去程不适用日期, 1 回程不适用日期)
TimeS	String	是	不适用日期, 开始日期
TimeE	String	是	不适用日期, 结束日期
乘机人类型 PType Element			
字段名称	类型	是否必填	描述
Type	String	是	乘机人类型 (ADU=成人, CHD=儿童, INF=婴儿)
Discount	String	是	代理费, 数字
Agio	String	是	奖励, 数字
AddMoney	String	是	开票费, 数字
Fare	String	是	票面价
Tax	String	是	税费
TotalPrice	String	是	结算价 (不含平台管理费), 数字

### 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>testrafree</Partner>
  <Action>GetPolicyByPnrText</Action>
  <RequestFmt>XML</RequestFmt>
  <ResponseFmt>XML</ResponseFmt>
  <SignType>MD5</SignType>
  <Sign>3f4c306587fc5a271c8c59e395e87190</Sign>
  <Version>1.0.0.0</Version>
  <Param>
    <UserName>testrafree</UserName>
    <ChangePnr>false</ChangePnr> //按需求设置 true/false
  </Param>
</Request>
```

```
<Category>false</Category> //按需求设置 true/false
<NoRMB>false</NoRMB> //按需求设置 true/false
<IsGetSingle>false</IsGetSingle> //按需求设置 true/false
<RequirePolicyCount>2</RequirePolicyCount> // IsGetSingle为false时此项设置才有效
<LowerWaitTime>10</LowerWaitTime> //按需求设置 0-50
<PnrContent>52544B4E475339380D0A20312E57554D414945522F4D494A495449204D5220204B4E47533938
0A20322E202043413134373820543120204D4F30394D4159202055524350454B20484B3120202031393135203232
35302020202020202020204520543254330A20332E20204341383337202054312020545531304D415920205045
4B484E4C20484B312020203031353020313733302D3120202020202020452054332D2D0A20342E202043413833
38202056312020534130394A554C2020484E4C50454B20484B312020203233333020303435352B322020202020
202045202D2D54330A20352E202043413139303120563120204D4F31314A554C202050454B55524320484B312020
2030393330203133343020202020202020204520543354320A20362E434B472F5420434B472F54203032332D
36333630353931382F434B47204C49414E2046454E4720414952205345525649434520434F2E2C4C54442F4C4955
2042494E2041473A31393339302F20202020382F3430302F322F3320414243444546470A20372E31333737343239
333838332D53595354454D0A20382E3032312D35313036393939392F50310A20392E303531332D38393138363636
362F50310A31302E544C2F313230302F30374D415931362F434B473231360A31312E535352204F54485320434120
4641494C454420544F2055534520464C4947485420434F55504F4E5320494E2053455155454E43452057494C4C20
524553554C542050454E414C545920414E4420204144444954494F4E414C2046415245205041594D454E540A3132
2E535352204144544B20314520425920434B47313041505231362F31303237204F522043584C20434120414C4C20
534547530A31332E53535220444F435320434120484B3120502F434E2F4736303434393534322F434E2F31354155
4734322F4D2F30374150523232F57554D414945522F4D494A4954492F50310A31342E4F53492043412043544354
31333737343239333838330A31352E524D4B20544A2041555448205348413731370A31362E524D4B2043412F4D42
473742310A31372E434B473231360A</PnrContent>
</Param>
</Request>
```

生成的待签名的字符串:

```
Action=GetPolicyByPnrText&Param=Category=false&ChangePnr=false&IsGetSingle=false
&NoRMB=false&PnrContent=52544B4E475339380D0A20312E57554D414945522F4D494A4954492
04D5220204B4E475339380A20322E202043413134373820543120204D4F30394D415920205552
4350454B20484B312020203139313520323235302020202020202020204520543254330A2033
2E20204341383337202054312020545531304D4159202050454B484E4C20484B31202020303135
3020313733302D3120202020202020452054332D2D0A20342E2020434138333820205631202
0534130394A554C2020484E4C50454B20484B312020203233333020303435352B322020202020
20202045202D2D54330A20352E202043413139303120563120204D4F31314A554C202050454B5
5524320484B312020203039333020313334302020202020202020204520543354320A20362E4
34B472F5420434B472F54203032332D36333630353931382F434B47204C49414E2046454E47204
14952205345525649434520434F2E2C4C54442F4C49552042494E2041473A31393339302F20202
020382F3430302F322F3320414243444546470A20372E31333737343239333838332D535953544
54D0A20382E3032312D35313036393939392F50310A20392E303531332D38393138363636362F
50310A31302E544C2F313230302F30374D415931362F434B473231360A31312E535352204F5448
53204341204641494C454420544F2055534520464C4947485420434F55504F4E5320494E205345
5155454E43452057494C4C20524553554C542050454E414C545920414E4420204144444954494F
```

4E414C2046415245205041594D454E540A31322E535352204144544B20314520425920434B473  
13041505231362F31303237204F522043584C20434120414C4C20534547530A31332E535352204  
44F435320434120484B3120502F434E2F4736303434393534322F434E2F313541554734322F4D2  
F303741505232332F57554D414945522F4D494A4954492F50310A31342E4F53492043412043544  
35431333737343239333838330A31352E524D4B20544A2041555448205348413731370A31362E  
524D4B2043412F4D42473742310A31372E434B473231360A&UserName=testrafree&Partner=testr  
afree&RequestFmt=XML&ResponseFmt=XML&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:

失败示例:

```
<?xml version="1.0"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageModel>
    <MessageCode>CallFail</MessageCode>
    <Description>没有符合条件的政策</Description>
  </MessageModel>
</Response>
```

成功示例:

```
<?xml version="1.0" ?>
<Response>
  <MessageModel>
    <MessageCode>CallSuccess</MessageCode>
    <Description>查询成功</Description>
  </MessageModel>
  <Price>
    <Type>0</Type>
    <Fare>6860.00</Fare>
    <Tax>2849</Tax>
    <Total>9709</Total>
    <IsQPoint>false</IsQPoint>
    <Qpoint>0</Qpoint>
    <IT>false</IT>
    <NetFare>false</NetFare>
    <Commision>0.00</Commision>
  </Price>
  <PriceList /> <!--儿童和婴儿价格信息 显示在此节点里-->
<PnrFlightList>
  <PnrFlightInfo>
```

```

<AirCompany>MU</AirCompany>
<Start>BKK</Start>
<End>KMG</End>
<StartCity>BKK</StartCity>
<EndCity>KMG</EndCity>
<FareBasic>NKS RSA</FareBasic>
<FareBasicFSU />
<Nubmer>742</Nubmer>
<FlightDate>2018-08-28</FlightDate>
<FlightFTIME>15:45</FlightFTIME>
<FlightTTIME>19:10</FlightTTIME>
<Cab>R</Cab>
<CabChild />
<FlightState>HK1</FlightState>
<ShareCompany />
<IsShare>>false</IsShare>
<IsOpen>>false</IsOpen>
<IsMain>1</IsMain>
<Type>0</Type>
<luggage>2PC</luggage>
</PnrFlightInfo>
</PnrFlightList>
= <PolicyList>
= <PolicyInfo>
<id>/0U/+CGK9GU=</id>
<ed>8F1B89D873397647FE896AF7F720237C1257CBC8D88527630323E2C
    BACE4E80C12E183EA3F9ABE284DAD7C4E3BFAD497</ed>
<Airline>CA</Airline>
<DepCity>CN</DepCity>
<ArrCity>AU/NZ/FJ/CK/GU/KI/NC/PF/PG/PW/SB/TO/TV/VU/WS/BE/GB
    /LU/NL/FR/IE/MC/EE/LT/LV/RU/BY/MD/UA/DK/FI/NO/SE/IS/ES/GR
    /IT/MT/PT/TR/BG/CY/HR/RO/SI/AL/BA/ME/MK/RS/DE/CH/AT/CZ/H
    U/PL/SK/US/CA/CR/GT/HN/MX/NI/PA/SV/AG/KY/TC/DM/GP/HT/PR/
    BB/CU/BZ/BS/BM/JM/AW/GY/</ArrCity>
<SaleSDate>2016-04-01</SaleSDate>
<SaleEDate>2016-12-31</SaleEDate>
<Discount>0</Discount>
<Cabin>A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z/
    </Cabin>
<Agio>6.9</Agio>
<AddMoney>0</AddMoney>
<ProviderType>1</ProviderType>
<CurrencyCode>CNY</CurrencyCode>
<RateValue>1</RateValue>
    
```

```

<FlightType>0/1</FlightType>
<TicketS>2016-04-01</TicketS>
<TicketE>2016-12-31</TicketE>
<Remark>[票本类型]: 境外电子、<br/>[儿童票]: 奖励与成人一致 <br/>[婴儿票]: 可
  开无奖励 <br/>[PNR]: 换编码出票<br/>[不适用]: 留学生、移民、劳工、海员、青年、
  老年、探亲、OPEN、SPA、三方协议、套票、整团、小团、<br/>换编不授权, <br/>[其
  他备注]: 境外配置出票, 退改规定以黑屏为准; 下午 3 点开票当日 20:00 点前可以废票, 周
  一至周日都可以办理退改, 退票 24 小时内退回票款, 无法提供行程单。代码共享、联运不支
  持<br/></Remark>
<eReturntrip>2017-01-31</eReturntrip>
<officeid>SHA327</officeid>
<isCHangePnr>A</isCHangePnr>
<UpdateDate>1459992643</UpdateDate>
<Z>1</Z>
<ZDepCity>A00/</ZDepCity>
  <ZCabin>A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z
  /</ZCabin>
<IsCategory>C</IsCategory>
<IsWaste />
<IsChildren>A</IsChildren>
<IsBaby>A</IsBaby>
<IsShare />
<IsAdapt>A,B,C,D,E,R,F,N,P,H,I,J,K</IsAdapt>
<Price>1500</Price>
<isreward>A</isreward>
<IsAuto>0</IsAuto>
<AdvanceDays>2</AdvanceDays>
<pbeftime>08:01</pbeftime>
<pendtime>20:00</pendtime>
<pbeftime_>00:00</pbeftime_>
<pendtime_>00:00</pendtime_>
<pbeftime_i>00:00</pbeftime_i>
<pendtime_i>00:00</pendtime_i>
<OneSubtracting>0</OneSubtracting>
<RoundSubtracting>0</RoundSubtracting>
<ChildrenSingle>0</ChildrenSingle>
<ChildrenRate>0.0</ChildrenRate>
<ChildrenFee>0</ChildrenFee>
<BabyFee>0</BabyFee>
<Apply_AgeStart>0</Apply_AgeStart>
<Apply_AgeEnd>0</Apply_AgeEnd>
<Apply_People>0</Apply_People>
<OnlyPeopleEven>0</OnlyPeopleEven>
<ChangePnrAuthorize>1</ChangePnrAuthorize>
  
```



```
<NoOutTicketOffice />
<FareBasisCodeInclude_NOClose>0</FareBasisCodeInclude_NOClose>
<FareBasisCodeInclude />
<ITPriceNoReward>0</ITPriceNoReward>
<ITPriceMoreAddFee>0</ITPriceMoreAddFee>
<FareBasisCode>0</FareBasisCode>
<BelowPar>0</BelowPar>
<NoOutTicketOfficeInfo>0</NoOutTicketOfficeInfo>
<RoundMustSpace />
<MustProviderReview>0</MustProviderReview>
<NotApplicableAirLine />
<RetrunAirportType>1</RetrunAirportType>
<RetrunAirport />
<TourCode />
<TransitAirport />
<ExFlights />
<UnTimes />
= <PTs>
  <PType Type="ADU" Fare="6860" Tax="2849" Discount="0" Agio="6.9"
    AddMoney="0" TotalPrice="9236" />
  <PType Type="CHD" Fare="6860" Tax="2849" Discount="0" Agio="6.9"
    AddMoney="0" TotalPrice="9236" />
</PTs>
<Effective>2</Effective>
</PolicyInfo>
</PolicyList>
</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.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 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
}
```