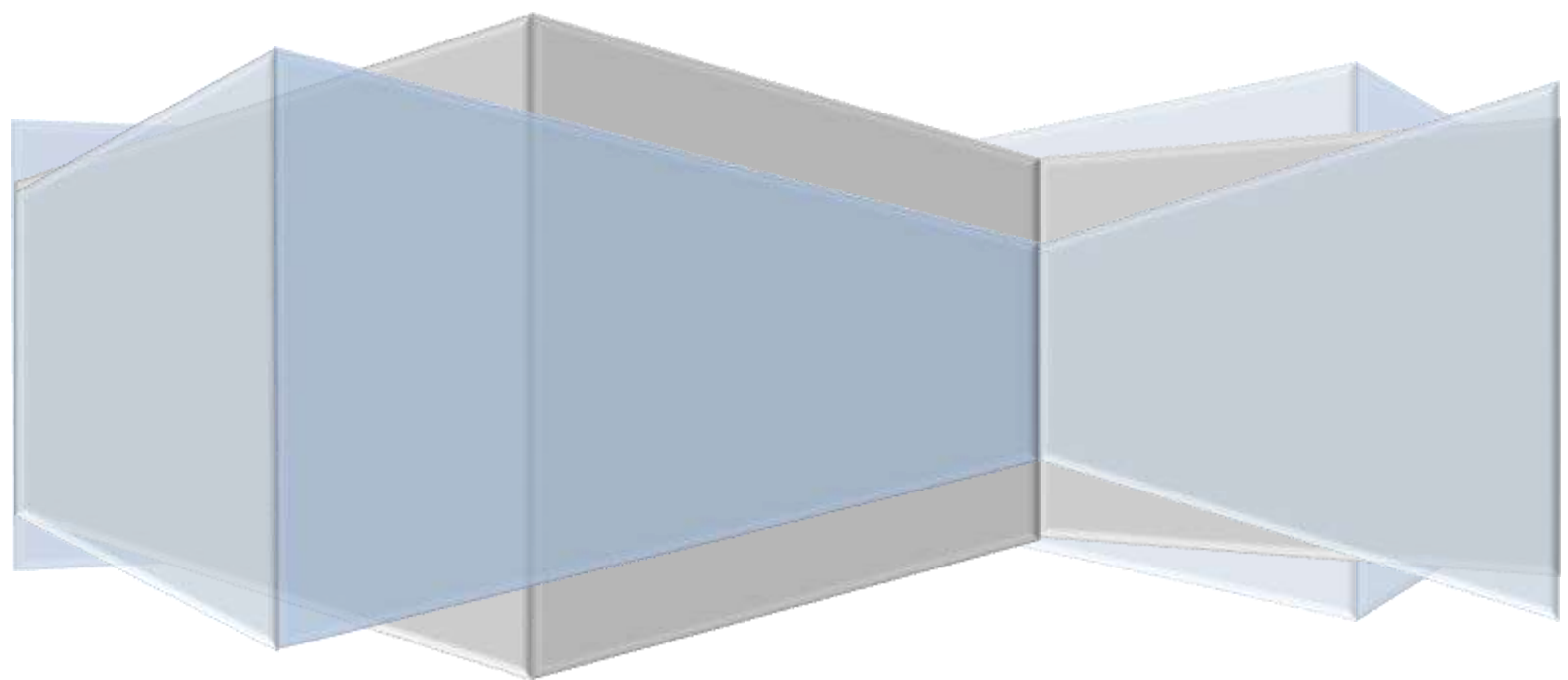




同程旅游

同程机票外部运价接口

v1.4



目录

同程机票外部运价接口 v1.4.....	0
1 签名规则.....	2
2 系统交互时序图.....	3
3 功能接口说明.....	4
3.1 外部运价推送接口.....	4
3.1.1 接口地址.....	4
3.1.2 输入项.....	4
3.1.3 输出项.....	8
3.2 外部运价清除接口.....	8
3.2.1 接口地址.....	8
3.2.2 输入项.....	8
3.2.3 输出项.....	9
3.3 下单接口.....	10
3.3.1 输入项.....	10
3.3.2 输出项.....	12
3.4 支付前校验.....	12
3.4.1 输入项.....	12
3.4.2 输出项.....	13
3.5 通知出票.....	13
3.5.1 输入项.....	13
3.5.2 输出项.....	14
3.6 客票回填接口.....	14
3.6.1 接口地址.....	14
3.6.2 输入项.....	14
3.6.3 输出项.....	15
3.6.4 枚举说明.....	16
3.7 订单取消通知接口.....	20
3.7.1 输入项.....	20
3.5.2 输出项.....	21

1 签名规则

所有的通讯都采用 HTTP 协议，请求动词均为 POST，请求的参数以 JSON 格式提交，提交编码为 UTF-8。HTTP HEADER 中的“Content-type”字段要设置值“application/json”。在 HTTP HEADER 中需要添加以下自定义值头，以方便平台对供应商身份校验：

HTTP HEADER	值	说明
X-MERCHANT-ID	76344889	由同程分配的供应商唯一的身份 ID
X-TIMESTAMP	1460534526137	请求时间戳毫秒
X-SIGNDATA	本次请求的签名	本次请求的签名，算法见下

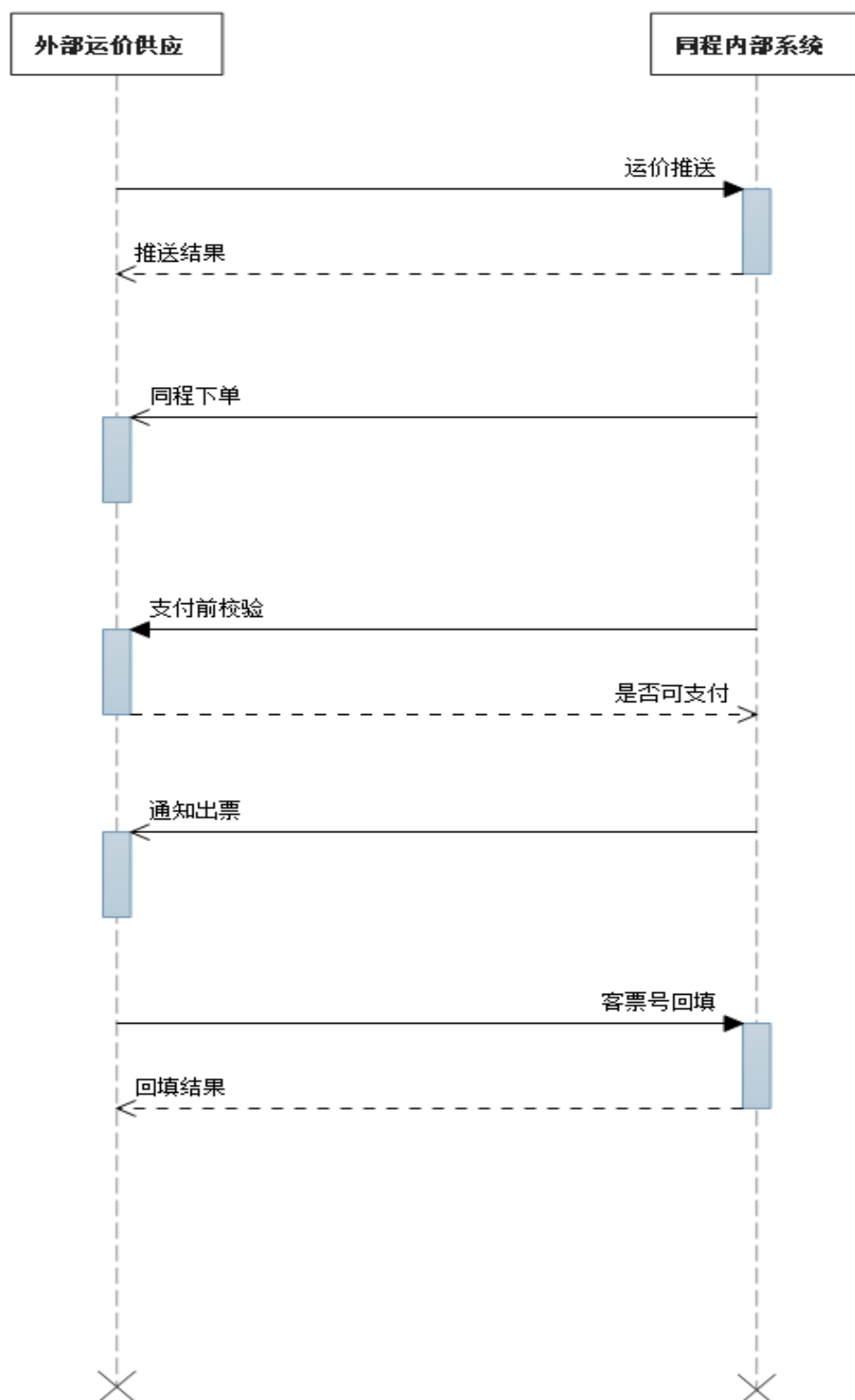
在同程的供应商平台注册成为供应商后，将会被分配到唯一的身份 ID 和身份校验码（token），其中身份 ID 将做为 HEADER 中的 X-MERCHANT-ID 值传递，token 将做为签名的密钥，供应商需要妥善保管该 token。签名为对 X-MERCHANT-ID+token+ X-TIMESTAMP 的 MD5 摘要（小写），签名值将做 X-SIGNDATA，传递给同程，同程会根据签名规则再次校验签名，若校验不过，则拒绝本次请求。

C#代码示例：

```
/// <summary>
/// 时间帮助类
/// </summary>
public class TimeHelper
{
    /// <summary>
    /// 获取当前协调时间戳
    /// </summary>
    /// <returns></returns>
    public long GetTimeStampUtcNow()
    {
        TimeSpan timeSpan = DateTime.UtcNow - new DateTime(1970, 1, 1);
        return Convert.ToInt64(timeSpan.TotalMilliseconds);
    }
}

//签名为对 X-MERCHANT-ID+token+ X-TIMESTAMP 的 MD5 摘要（小写）
string publicSign =
System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(publicKey,
"MD5").ToLower();
```

2 系统交互时序图



3 功能接口说明

3.1 外部运价推送接口

供应商将航司运价推送给同程 OpenAPI 外部运价推送接收接口。

接口调用方：供应商

接口提供方：同程

使用说明：供应商获取最新的航司运价后按照标准格式推送给同程的平台。

过期时间：如果不使用清除接口下架航线运价，目前默认 400 小时后过期。

存储维度：SZX^HGH^2016-12-10^OW^ZH^76345029 我们按照此格式存储您推送的运价

3.1.1 接口地址

请求 URI: <http://tcflightopenapi.17uoft.com/ExternalPrice/PricePush.ashx>

3.1.2 输入项

请求参数：

字段名	类型	是否必须	说明
supplierId	String	是	供应商 ID
isDelByFlightNos	Int	否	删除数据是否精确到航班号纬度 (默认)1:是 ; 0:否)
flightSegmentList	数据集合 List<FlightSegment >	是	航段集合

FlightSegment:

字段名	类型	是否必须	说明
airlineCode	String	是	航司二字码：CA
originCity	String	是	起飞城市 3 字码
destinationCity	String	是	到达城市 3 字码
flightDate	String	是	航班日期 “yyyy-MM-dd”
tripType	String	是	航段类型： 单程 OW 往返 RN 目前只支持 单程的
flightList	数据集合	是	航班运价数据集合



	List<Flight>		
--	--------------	--	--

Flight:

字段名	类型	是否必须	说明
flightNo	字符串	是	航班号：CA5741
departureTime	字符串	是	起飞时间 “yyyy-MM-dd HH:mm:ss ”
arriveTime	字符串	是	到达时间：“yyyy-MM-dd HH:mm:ss ”
stops	整型	是	经停次数 0:无经停 1:经停一次 2:经停二次
airCraftStyle	字符串	是	机型
baseFare	浮点型数据	是	基准舱运价
cabinList	数据集合 List<Cabin>	是	舱位数据集合

Cabin:

字段名	类型	是否必须	说明
cabinCode	字符串	是	舱位代码：“Y”
cabinName	字符串	否	舱位名称：“经济舱 ”
inventory	整形	是	可销售数量 1...9 大于 9 则统一为 10
productList	数据集合 List<Product>	是	产品集合

Product:

字段名	类型	是否必须	说明
productId	字符串	是	产品 ID 如果没有则传值 0
productCode	字符串	是	产品 code 如果没有则传空字符串
productName	字符串	是	产品名称 如果没有则传空字符串
adultFare	FareModel	是	成人运价
childFare	FareModel	否	儿童运价 如果不做儿童运价销售则可传不传

FareModel:

字段名	类型	是否必须	说明
farePrice	浮点型数据	是	销售价格 单位元
marketFare	浮点型数据	是	票面价格 单位元
airportTax	浮点型数据	是	机场建设费 单位元
fuelTax	浮点型数据	是	燃油税费 单位元 没有则传 0.00
otherTax	浮点型数据	是	其他费用 单位元 没有则传 0.00

入参 Json 示例:

```
{
  "supplierId": "76345102",
  "isDelByFlightNos ": 1,
  "flightSegmentList": [
    {
      "airlineCode": "ZH",
      "originCity": "SZX",
      "destinationCity": "XIY",
      "flightDate": "2016-08-01",
      "tripType": "OW",
      "flightList": [
        {
          "flightNo": "ZH9241",
          "airCraftStyle": "32F",
          "departureTime": "",
          "arriveTime": "2016-08-01 10:15:00",
          "stops": 0,
          "baseFare": 1640.00,
          "cabinList": [
            {
              "cabinCode": "F",
              "cabinName": "头等舱",
              "inventory": 6,
              "productList": [
                {
                  "productId": "",
                  "productCode": "ZH9241F",
                  "productName": "",
                  "adultFare": {
                    "farePrice": 3050.00,
                    "marketFare": 3280.00,
                    "airportTax": 50.00,
                    "fuelTax": 0.00,
                    "otherTax": 0.00
                  },
                  "childFare": null
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```


3.1.3 输出项

响应参数:

字段名	字段值	说明
code	字符串	响应 code
message	字符串	响应消息

成功出参 Json 示例:

```
{
  "code": "success",
  "message": "推送成功"
}
```

失败出参 Json 示例:

```
{
  "code": "failure",
  "message": "推送失败"
}
```

3.2 外部运价清除接口

供应商需要下线某些航线的外部运价，可以访问外部运价推送清除接口。

接口调用方: 供应商

接口提供方: 同程

使用说明: 用于主动无效某些航线运价。

3.2.1 接口地址

请求 URI: <http://tcflightopenapi.17usoft.com/ExternalPrice/PriceClear.ashx>

3.2.2 输入项

请求参数:

字段名	类型	是否必须	说明
supplierId	字符串	是	供应商 ID
flightSegmentClearList	数据集 List<FlightSegment >	是	航段集合

flightSegmentClear:

字段名	类型	是否必须	说明
airlineCode	字符串	是	航司二字码：CA
originCity	字符串	是	起飞城市 3 字码
destinationCity	字符串	是	到达城市 3 字码
flightDate	字符串	是	航班日期 “yyyy-MM-dd”
tripType	字符串	是	航段类型： 单程 OW 往返 RN 目前只支持 单程的

入参 Json 示例:

```
{
  "supplierId": "76345102",
  "flightSegmentClearList": [
    {
      "airlineCode": "ZH",
      "originCity": "SZX",
      "destinationCity": "XIY",
      "flightDate": "2016-08-01",
      "tripType": "OW"
    },
    {
      "airlineCode": "ZH",
      "originCity": "SZX",
      "destinationCity": "SHA",
      "flightDate": "2016-08-01",
      "tripType": "OW"
    }
  ]
}
```

3.2.3 输出项

响应参数:

字段名	字段值	说明
code	字符串	响应 code
message	字符串	响应消息

成功出参 Json 示例:

```
{
  "code": "success",
```

```
"message": "清除成功"
}
```

失败出参 Json 示例:

```
{
  "code": "failure",
  "message": "清除失败"
}
```

3.3 下单接口

旅客发起下单后, 同程调用此接口向供应商发起下单确认, 供应商根据订单参数去航司网站进行实时占座或异步占座。

接口提供方: 供应商

接口调用方: 同程

请求 URL: 供应商提供

使用说明: 供应商下单接口耗时应在 10s 内, 所以建议供应商进行异步占座

3.3.1 输入项

入参:

字段名	类型	是否必须	说明
tcOrderNo	字符串	是	同程平台订单流水
orderAmount	浮点型数据	是	订单总金额 1500.00
depCode	字符串	是	出港城市三字码 "PEK"
arrCode	字符串	是	到港城市三字码 "HAK"
flightDate	字符串	是	航班日期 "yyyy-MM-dd"
flightNo	字符串	是	航班号 "ZH9999"
airlineCode	字符串	是	航司二字码 "ZH"
cabinCode	字符串	是	舱位代码 "C"
productCode	字符串	是	产品代码 "JJX"
passengerInfo	数据集合 List<Passenger>	是	乘客信息集合
orderCreateTime	字符串	是	平台订单创建时间 "yyyy-MM-dd HH:mm:ss"

Passenger:

字段名	类型	是否必须	说明
passengerId	字符串	是	同程平台乘客 ID



passengerName	字符串	是	乘客姓名
passengerType	字符串	是	乘客类型 成人：ADULT 儿童：CHILD 婴儿：INFANT
birthday	字符串	否	旅客生日 如果乘客类型为儿童则必传 “yyyy-MM-dd”
certNo	字符串	是	旅客证件号
certType	字符串	是	旅客证件类型： NI:身份证 JG:军官证 ID:其它 PP:护照
fareInfo	FareModel	是	旅客运价信息

FareModel:

字段名	类型	是否必须	说明
baseFare	浮点型数据	是	销售价格 单位元 620.00
marketFare	浮点型数据	是	票面价格 单位元 620.00
airportTax	浮点型数据	是	机场建设费 单位元 50.00
fuelTax	浮点型数据	是	燃油税费 单位元 没有则传 0.00
otherTax	浮点型数据	是	其他费用 单位元 没有则传 0.00

入参 Json 示例:

```
{
  "airlineCode": "ZH",
  "arrCode": "HAK",
  "cabinCode": "A",
  "depCode": "SZX",
  "flightDate": "2016-07-23",
  "flightNo": "ZH9909",
  "orderAmount": 720,
  "orderCreateTime": "2016-05-20 10:00:00",
  "passengerInfos": [
    {
      "birthday": "1981-08-05",
      "certNo": "110101198108054136",
      "certType": "NI",
      "fareInfo": {
        "airportTax": 50,
        "baseFare": 670,
        "fuelTax": 0,
```

```
        "marketFare": 700,
        "otherTax": 0
    },
    "passengerId": "0001",
    "passengerName": "独孤封",
    "passengerType": "ADULT"
}
],
"productCode": "ZH9909A",
"tcOrderNo": "FM201607010003"
}
```

3.3.2 输出项

字段名	类型	是否必须	说明
code	字符串	是	接口响应消息 code “0” 为成功 其他为失败
message	字符串	是	接口响应消息 成功可传空字符串，失败响应具体处理失败信息
result	Object	否	返回信息

Result 实体

字段名	类型	是否必须	说明
orderNo	字符串	否	供应商订单流水号（长度在 50 个字符内），供应商确认请求数据可下单，则回传供应商订单流水号，

3.4 支付前校验

旅客下单成功后，发起支付前，同程调用此接口向供应商发起支付确认，以确保旅客支付后，供应商可以有效的出票。

接口提供方：供应商

接口调用方：同程

请求 URI：供应商提供

接口使用说明：供应商可根据供应商订单流水进行确认当前订单是否可继续支付并出票，接口响应耗时应在 10s 内

3.4.1 输入项

字段名	类型	是否必须	说明
-----	----	------	----



orderNo	字符串	是	供 应 商 订 单 流 水 号 : LH2016031300001
---------	-----	---	--------------------------------------

入参 Json 示例:

```
{  
  "orderNo": " LH2016031300001"  
}
```

3.4.2 输出项

字段名	类型	是否必须	说明
code	字符串	是	接口响应消息 code "0" 为成功 其它均为失败
message	字符串	是	接口响应消息,成功可传空字符串,失败回写具体消息

出参 Json 示例:

```
{  
  "code": "0",  
  "message": ""  
}
```

3.5 通知出票

旅客支付成功后,同程会调用此接口向供应商发起请求出票,供应商对订单进行支付出票。

接口提供方: 供应商

接口调用方: 同程

接口 URI: 供应商提供

接口使用说明: 供应商收到出票请求后,若确认当前订单已无法给同程出票,则回写失败信息给同程,同程可从其它渠道给旅客出票,以减少同程的损失。接口响应时长应在 10s 内。

3.5.1 输入项

字段名	类型	是否必须	说明
orderNo	字符串	是	供 应 商 订 单 流 水 号 : LH2016031300001

入参 Json 示例:

```
{
```

```
"orderNo": " LH2016031300001"
}
```

3.5.2 输出项

字段名	类型	是否必须	说明
code	字符串	是	接口响应消息 code “0” 为成功 其它均为失败
message	字符串	是	接口响应消息,成功可传空字符串,失败回写具体消息

出参 Json 示例:

```
{
  "code": "0",
  "message": ""
}
```

3.6 客票回填接口

供应商出票之后将客票号信息同步至同程

接口提供方: 同程

接口调用方: 供应商

接口使用说明: 供应在出票后将出票信息同步给同程, 同程内部会对该信息进行一定的校验, 校验失败的回填信息会被驳回。

3.6.1 接口地址

同程客票回填接口地址: <http://tcflightopenapi.17usoft.com/tc/ticketnotify.ashx>

3.6.2 输入项

字段名	类型	是否必须	说明
OrderSerialid	字符串	是	订单流水号
IsTicketSuccess	字符串	是	出票状态
Username	字符串	是	用户名
Password	字符串	是	密码格式
Remark	字符串	否	出票备注
DifferencePrice	浮点型数值	否	差价
ticketInfo	TicketInfo	是	客票信息

TicketInfo:

字段名	类型	是否必须	说明
PassengerName	字符串	是	乘客姓名
Pnr	字符串	是	出票 PNR，如换编码则填新编码
TicketNo	字符串	是	13 位票号 例：8471952031274

密码加密方式:

```
System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile("517Na" + "#" + "517Na", "MD5").ToLower();
```

账号加密示例:

加密前:

账号: tc

密码: tc#tc@456

加密后:

账号: tc

密码: 76adef40d7b693da68ba2c0cb6512fb9

入参 Json 示例:

```
{
  "OrderSerialid": "FP54193AC23A83801392",
  "IsTicketSuccess": "T",
  "Remark": "机建费应该是 100",
  "DifferencePrice": "123.45",
  "ticketInfo": [
    {
      "PassengerName": "杨进龙",
      "Pnr": "N45PDL",
      "TicketNo": "2302569874563"
    }
  ],
  "Username": "517na",
  "Password": "8139cfb7818d54b957f768b0a62ddf9d"
}
```

3.6.3 输出项

字段名	类型	是否必须	说明
-----	----	------	----

ErrorCode	字符串	是	接口响应码
ErrorMsg	字符串	是	接口响应消息(详见 枚举说明)

出参 Json 示例:

```
{
  "ErrorCode": "10",
  "ErrorMsg": "HASTICKETED"
}
```

3.6.4 枚举说明

```
/// <summary>
/// 成功
/// </summary>
[Description("成功")]
SUCCESS = 100000,
/// <summary>
/// 请求数据为空
/// </summary>
[Description("请求数据为空")]
NODATA = 100001,
/// <summary>
/// 数据格式错误
/// </summary>
[Description("数据格式错误")]
DATAFORMAT_INVALID = 100002,
/// <summary>
/// 用户名或者密码错误
/// </summary>
[Description("用户名或者密码错误")]
USERNAME_OR_PASSWORD_ERROR = 100003,
/// <summary>
/// 时间区间错误
/// </summary>
[Description("时间区间错误")]
DATERANGE_ERROR = 100004,
/// <summary>
/// 无效的订单状态
/// </summary>
[Description("无效的订单状态")]
ORDERSTATUS_INVALID = 100005,
/// <summary>
```

```
/// 订单列表为空
/// </summary>
[Description("订单列表为空")]
ORDERLIST_EMPTY = 100006,
/// <summary>
/// 无效订单号
/// </summary>
[Description("无效订单号")]
ORDERDETAIL_IS_NULL = 100007,
/// <summary>
/// 出票信息错误
/// </summary>
[Description("出票信息错误")]
TICKETINFO_ERROR = 100008,
/// <summary>
/// 出票信息乘客姓名错误
/// </summary>
[Description("出票信息乘客姓名错误")]
TICKETINFO_NAMEERROR = 100009,
/// <summary>
/// 该订单已出票，请勿重复提交
/// </summary>
[Description("该订单已出票，请勿重复提交")]
HASTICKETED = 100010,
/// <summary>
/// 该订单已退票，无法通知出票
/// </summary>
[Description("该订单已退票，无法通知出票")]
HASREFUNDED = 100011,
/// <summary>
/// 订单已确认
/// </summary>
[Description("订单已确认")]
HASCONFIRMED = 100012,
/// <summary>
/// 请求太频繁
/// </summary>
[Description("请求太频繁")]
REQUESTBUSY = 1000013,
/// <summary>
/// 用户不存在
/// </summary>
[Description("用户不存在")]
USESRNOTEXIST = 1000014,
```

```
/// <summary>
/// 出票回写所填人数不正确
/// </summary>
[Description("出票回写所填人数不正确")]
NUMBERERROR = 1000015,
/// <summary>
/// 客人未付款
/// </summary>
[Description("客人未付款")]
ClientPayERROR = 1000016,
/// <summary>
/// 没有权限
/// </summary>
[Description("没有权限执行操作")]
USERNOVERIFY = 1000017,
/// <summary>
/// 订单不存在
/// </summary>
[Description("订单不存在")]
ORDERNOTEXIST = 1000018,
/// <summary>
/// 变更单不存在
/// </summary>
[Description("变更单不存在")]
CHANGEORDERNOTEXIST = 1000019,
/// <summary>
/// 订单与变更单不匹配
/// </summary>
[Description("订单与变更单不匹配")]
CHANGEORDERNOMATCHING = 1000020,
/// <summary>
/// 无效变跟单号
/// </summary>
[Description("无效变更单号")]
CHANGEORDERDETAIL_IS_NULL = 1000021,
/// <summary>
/// 无效变跟单号
/// </summary>
[Description("变更单已被处理")]
CHANGEORDERHASPROCESS = 1000022,
/// <summary>
/// 退票金额验证失败
/// </summary>
```

```
[Description("退票金额验证失败")]
MONEYERROR = 1000023,
/// <summary>
/// 驳回理由为空
/// </summary>
[Description("驳回理由为空")]
REJECTREASONNULL = 1000024,
/// <summary>
/// 订单不能审核
/// </summary>
[Description("订单不能审核")]
ORDERCANNOTAUDIT = 1000025,
/// <summary>
/// 客票验证失败
/// </summary>
[Description("客票验证失败")]
TICKETCHECKERROR = 1000026,
/// <summary>
/// 客票验证失败，票号不存在
/// </summary>
[Description("客票验证失败，票号不存在")]
TICKETCHECKERROR_TICKETISNOT = 1000027,
/// <summary>
/// 客票验证失败，乘客姓名不正确
/// </summary>
[Description("客票验证失败，乘客姓名不正确")]
TICKETCHECKERROR_PASSENGERNAMEERROR = 1000028,
/// <summary>
/// 客票验证失败，客票号状态异常
/// </summary>
[Description("客票验证失败，客票号状态异常")]
TICKETCHECKERROR_TICKETERROR = 1000029,
/// <summary>
/// 客票验证失败，航班号不正确
/// </summary>
[Description("客票验证失败，航班号不正确")]
TICKETCHECKERROR_FLIGHTNOERROR = 1000030,
/// <summary>
/// 客票验证失败，起抵机场不符
/// </summary>
[Description("客票验证失败，起抵机场不符")]
TICKETCHECKERROR_STARTENDPORTERROR = 1000031,
/// <summary>
```

```

/// 客票验证失败，起飞时间不符
/// </summary>
[Description("客票验证失败，起飞时间不符")]
TICKETCHECKERROR_STARTENDTIMEERROR = 1000032,
/// <summary>
/// 客票验证中
/// </summary>
[Description("客票验证中，请稍后查询")]
TICKETCHECKERROR_VERIFICATING = 1000033,
/// 订单已驳回
/// </summary>
[Description("同程已提交申请驳回，请勿出票")]
TICKETSTATEERROR_REJECTED=1000034,
/// <summary>
/// 订单异常
/// </summary>
[Description("订单异常,请重新回填")]
TICKETERROR_ORDERINFOLOSE=1000035,
/// <summary>
/// <summary>
/// 未知系统异常
/// </summary>
[Description("未知系统异常")]
UNKNOWN_SYSTEM_ERROR = 101000

```

3.7 订单取消通知接口

旅客取消订单后，同程会调用此接口向供应商发起订单取消通知，供应商对订单进行取消操作以便释放舱位。

接口提供方：供应商

接口调用方：同程

接口 URI：供应商提供

接口使用说明：供应商订单取消通知后修改自己订单的状态，以便释放舱位，可以重新进新的订单。接口响应时长应在 10s 内。

3.7.1 输入项

字段名	类型	是否必须	说明
orderSerialId	字符串	是	同程订单流水号

入参 Json 示例：

```
{  
  "orderSerialId": "FS598A83C62100354859"  
}
```

3.5.2 输出项

字段名	类型	是否必须	说明
code	字符串	是	接口响应消息 code "0" 为成功 其它均为失败
message	字符串	是	接口响应消息,成功可传空字符串,失败回写具体消息

出参 Json 示例:

```
{  
  "code": "0",  
  "message": "成功"  
}
```