

**在线分销产品  
航班详细信息查询服务用户手册**

在线分销航班详细信息查询服务用户手册.....	1
1. 概述.....	3
2. 版本.....	3
3. 业务范围.....	4
4. 功能范围.....	4
5. 接口说明.....	4
5.1 接口结构.....	4
5.2 请求参数说明.....	8
5.3 响应参数说明.....	9
6. 错误说明.....	11
6.1http 协议错误.....	11
6.2 接口错误.....	11
7. 参考样例.....	12
7.1 以航班号+航空公司代码+出发地+目的地+舱位代码方式查询.....	12
7.2 以 PNR 号+航段序号方式进行查询.....	13
8. 接口调用.....	14
8.1 接口申请.....	14
8.2 接口认证.....	14
8.3 接口地址.....	14
8.4 JAVA 调用示例.....	14
8.5.NET 调用示例.....	17

# 1.概述

国内航班详细信息查询服务可以查询指定日期的航段上的航班具体信息，如航班的起飞降落城市，空中飞行时间，航班机型，餐食等。

# 2.版本

主要内容					
参考文档					
创建部门					
批准单					
版本号：					
批准人	批准人签字			批准日期	
文档控制					
版本	修改日期	修改内容描述	作者	批准人	批准日期
V1.0.0	2013/05/21	创建	李权		

### 3.业务范围

国内航班详细信息查询接口对应主机指令 DSG, 可以查看除座位可利用情况外的其他数据, 如航班的起飞降落城市, 起飞降落时间, 航班的空中飞行时间, 航班的空中飞行距离, 经停点数, 航班机型, 餐食等。

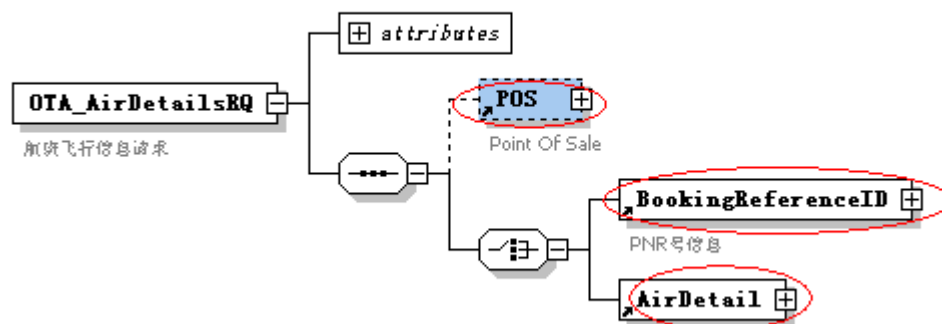
### 4.功能范围

- 支持以航班号, 舱位代码, 出发日期, 出发地, 目的地方式进行查询航班详细信息。
- 支持以 PNR 号和航段对应的序号方式进行航班信息查询

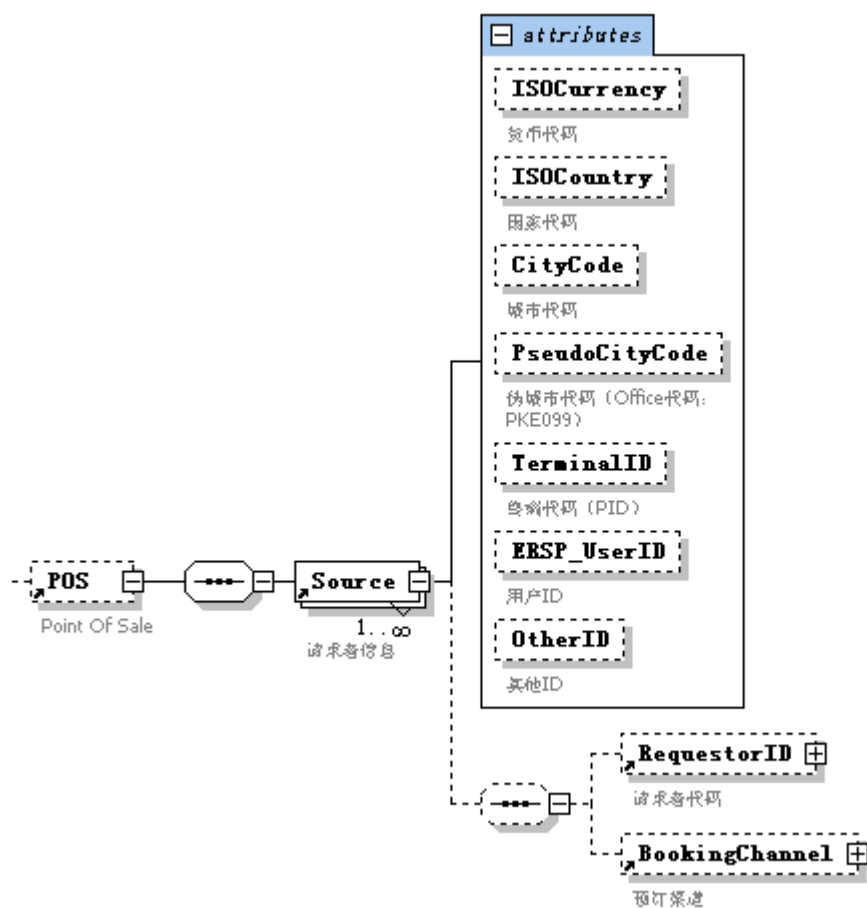
### 5.接口说明

#### 5.1 接口结构

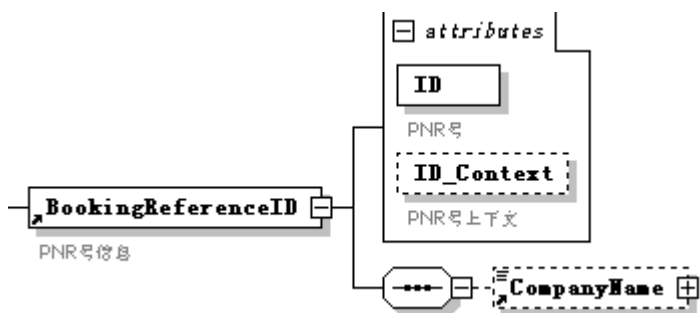
##### 5.1.1 请求结构



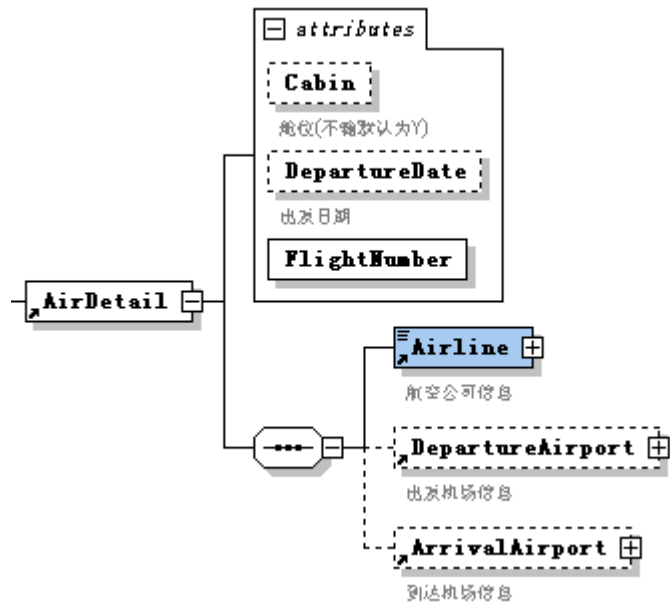
其中标识红色椭圆线的节点结构如下:  
POS 节点



其中 Source 节点表示零售商相关信息。  
BookReferenceID 节点

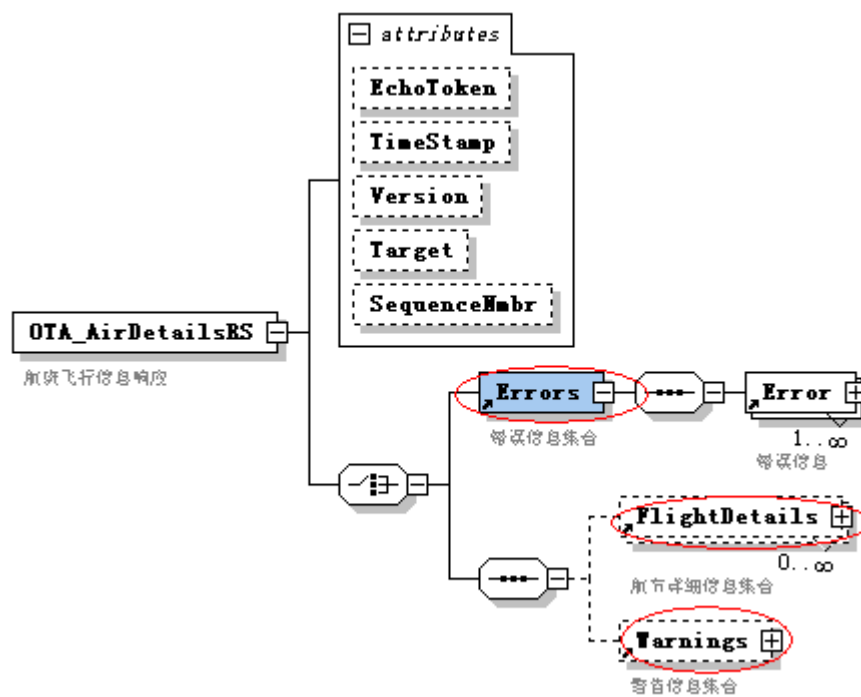


其中 ID 属性表示 PNR 号。  
AirDetail 节点

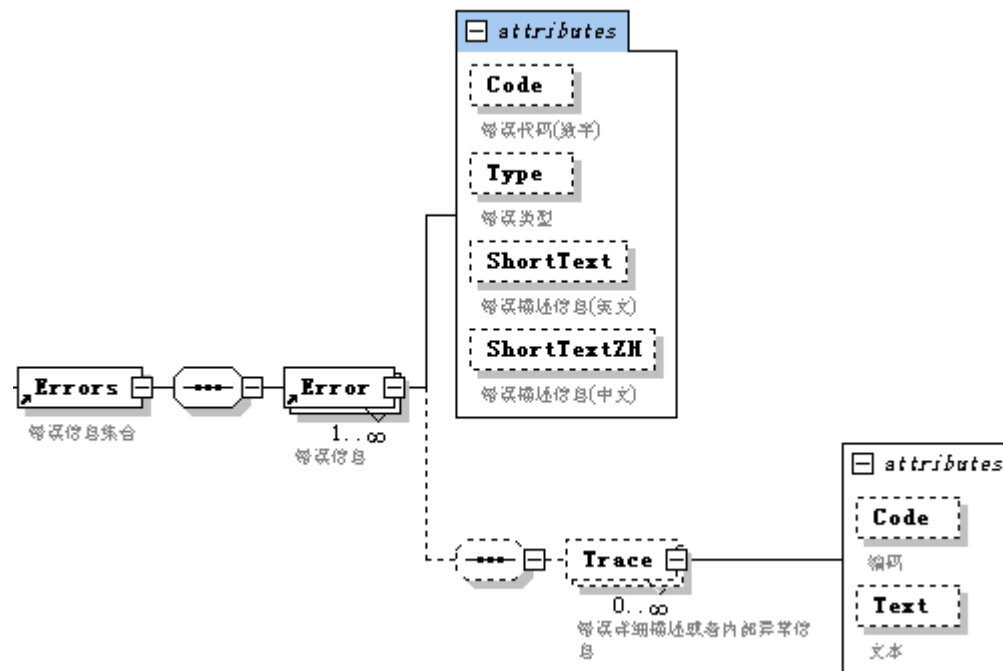


其中出发日期，航班号，航空公司为必填信息，舱位代码，出发机场信息，到达机场信息为选填信息。

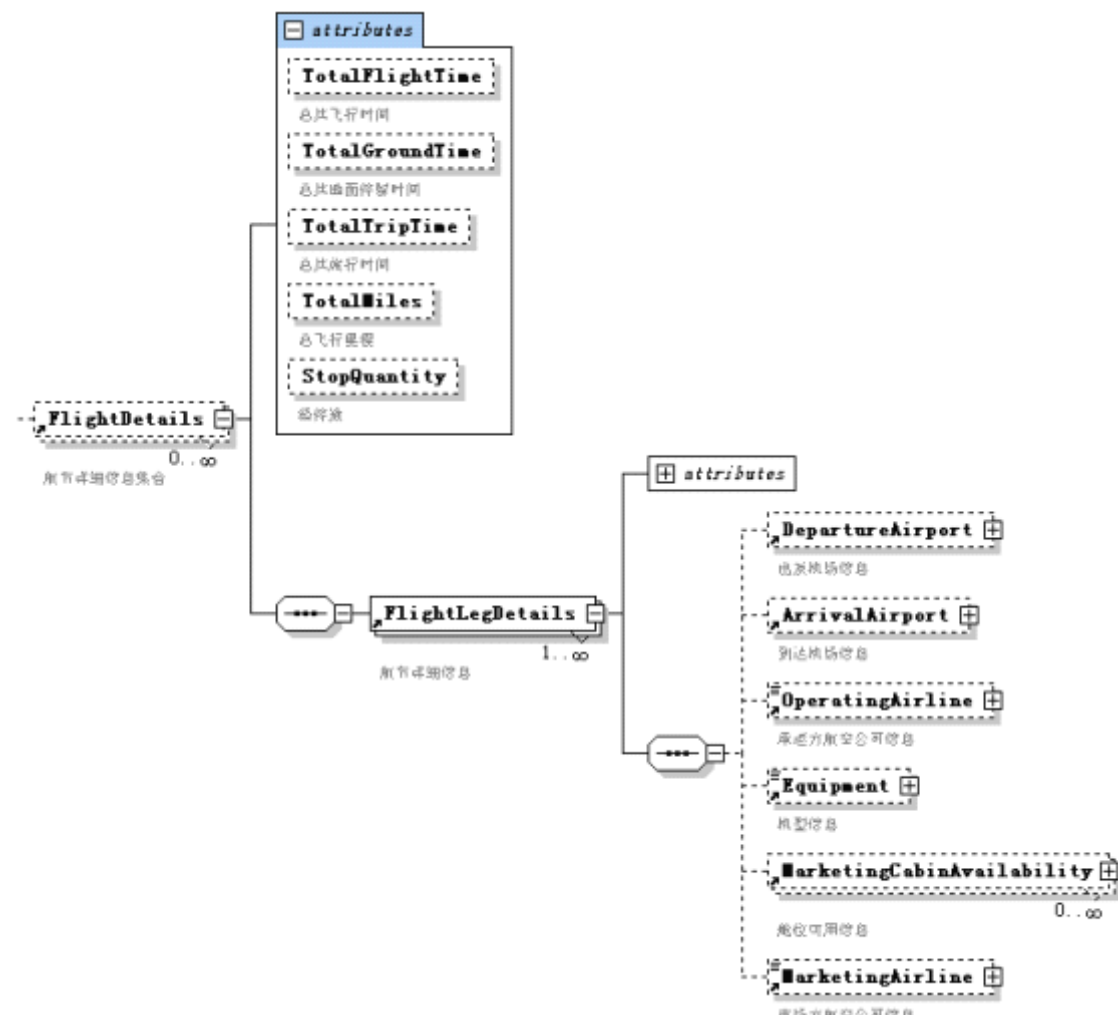
### 5.1.2 响应结构



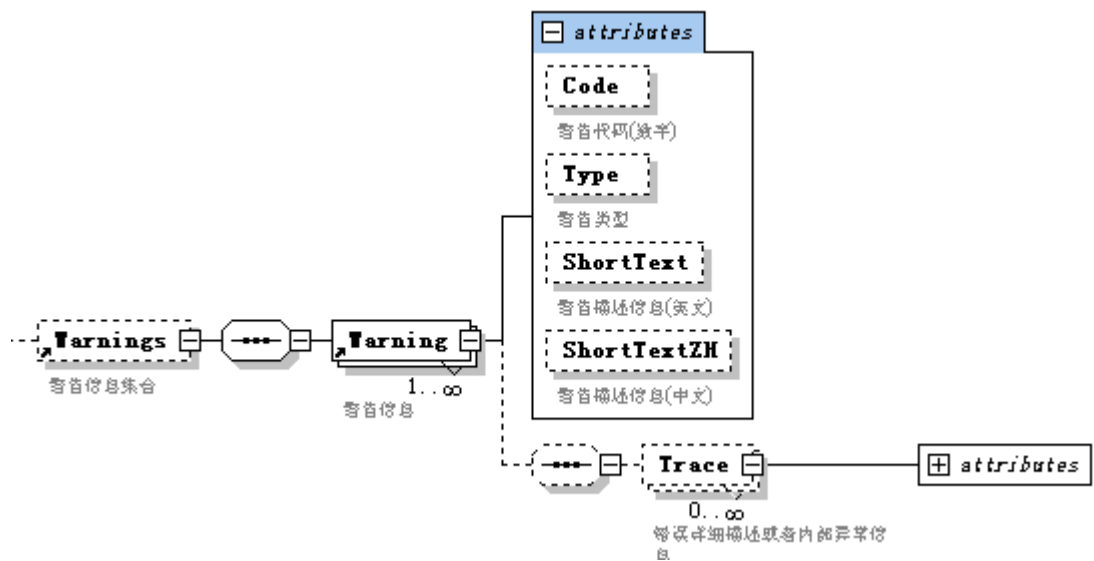
其中标识红色椭圆线的节点结构如下：  
Errors 节点



其中 Code 表示错误代码; Type 表示错误类型; ShortText 英文错误描述信息; ShortTextZH 中文错误信息描述, Trace 节点描述错误附加信息。  
FlightDetails 节点



其中 FlightLegDetails 是航段所有详细信息。  
Warning 节点



警告信息。

## 5.2 请求参数说明

元素名称	说明	Rep	类型	默认值	备注
OTA_AirDetailsRQ	航班详细信息 查询请求	1			
POS	零售商信息	0..1			
Source		1			
@PseudoCityCode	Office 号	0..1	string		
BookingReferenceID	PNR 信息	0..1			当以 PNR 号方式 查询时为必填项
@ID_Context	PNR 上下文	1	string		
@ID	PNR 信息	1	string		PNR 号
AirDetail	查询详细信息	0..1			当以航班号，出 发日期，航空公 司，出发机场， 目的机场方式查 询时必填项
@Cabin	舱位级别代码	0..1	string		
@DepartureDate	出发日期	0..1	date		格 式 如 ： 2013-05-09
@FlightNumber	航班号	1	string		
Airline	航空公司信息	1			
@Code	航空公司代码	0..1	string		
DepartureAirport	出发地信息	0..1			
@LocationCode	出发地 机场二	0..1	string		



	字码		
ArrivalAirport	目的地机场信息	0..1	
@LocationCode	目的地机场二字码	0..1	string

### 5.3 响应参数说明

元素名称	说明	Rep	类型	默认值	备注
OTA_AirDetailsRS	航班详细信息查询响应	1			
Errors					
Error	错误信息	0..1			
@Code	错误代码	0..1	string		
@Type	错误类型	0..1	string		
@ShortText	错误英文描述	0..1	string		
@ShortTextZH	错误中文描述	0..1	string		
Trace	错误附加信息	0..1			
@Code	错误附加信息代码	0..1	string		
@Text	错误附加信息描述	0..1	string		
FlightDetails		1			
@TotalFlightTime	总飞行时间	0..1	duration		
@TotalGroundTime	总共地面停留时间	0..1	duration		
@TotalTripTime	总旅行时间	0..1	duration		
@TotalMiles	总飞行里程	0..1	decimal		
@StopQuantity	经停数	0..1	decimal		
FlightLegDetails	航班里程信息	1..n			
@SequenceNumber	所有序号	0..1	decimal		
@FlightNumber	航班号	0..1	string		
@JourneyDuration	旅程时间	0..1	duration		
@GroundDuration	地面停留时间	0..1	duration		
@OnTimeRate	准点率	0..1	decimal		
@Ticket	票证类型	0..1		PAPER	Ticket 类型: PAPER 纸质票 ETICKET 电子客票 MCO
@DepartureDateTime	出发日期和时	0..1	dateTime		格式如:

	间			2001-12-17T09:30:47
@ArrivalDateTime	到达日期时间	0..1	dateTime	格式如: 2001-12-17T09:30:47
@DepartureDayOfWeek	出发日期是该周的周几	0..1		
@LegDistance	所有距离	0..1	decimal	
@DateChangeNbr	日期变更	0..1	string	
@SmokingAllowed	是否允许吸烟	0..1	boolean	
DepartureAirport	出发地机场信息	0..1		
@LocationCode	出发地机场二字码	0..1	string	
ArrivalAirportI	目的地机场信息	0..1		
@LocationCode	目的地机场二字码	0..1	string	
OperatingAirline	承运方航空公司信息	0..1		
@Code	承运方航空公司代码	0..1	string	
Equipment	机型信息	0..1		
@AirEquipType	飞机型号	0..1	string	
@ChangeofGauge	是否换过飞机	0..1	boolean	
MarketingCabinAvailability	舱位可用信息	0..1		
@CabinType	舱位类型	0..1	string	
Meal	餐饮信息	0..1		
@MealCode	餐饮代码	0..1	string	
MarketingAirline	市场方信息	0..1		
@Code	市场方二字码	0..1	string	
Warnings	警告信息集	0..1		
Warning	警告信息	1..n		
@Code	警告代码	0..1	string	
@Type	警告类型	0..1	string	
@ShortText	警告英文描述	0..1	string	
@ShortTextZH	警告中文描述	0..1	string	
Trace	警告附件信息	0..1		
@Code	警告附加信息代码	0..1	string	
@Text	警告附加信息描述	0..1	string	

## 6.错误说明

### 6.1http 协议错误

由于接口为 HTTP 接口，HTTP 协议会返回 403 错误：

http 错误代码	说明
403	用户名、密码、IP 错误或无权限

### 6.2 接口错误

<Errors>节点记录了错误信息。错误信息分为两类：业务错误、系统错误。详细见下表：

#### 系统错误

错误代码	错误描述	说明
-400	format of xml string is not valid	xml 内容格式错误
-402	input error	输入错误
-560	pnr locator is empty	pnr 号为空
-540	airline code is empty	航空公司代码为空
-541	flight number is empty	航班号为空
-510	departure datetime is empty	出发日期时间为空
-520	departure location code is empty	出发地代码为空
-521	arrival location code is empty	到达地代码为空
-10000	response is empty	返回信息为空

#### 业务错误

错误代码	错误描述	说明
-301	office prmission denied	没有 office 权限
-1	Biz Error	业务异常

## 7.参考样例

### 7.1 以航班号+航空公司代码+出发地+目的地+舱位代码方式查询

这种查询方式需要传入的参数包括：客户 office 号，航班号，航空公司二字码，出发地三字码，目的地三字码，舱位代码信息。其中，舱位代码信息可以为空，此时默认按 Y 舱进行处理。

请求示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirDetailsRQ>
  <POS>
    <Source PseudoCityCode="BJS191"/>
  </POS>
  <AirDetail DepartureDate="2013-06-10" FlightNumber="1531" Cabin="Y">
    <Airline Code="CA"/>
    <DepartureAirport LocationCode="PEK"/>
    <ArrivalAirport LocationCode="SHA"/>
  </AirDetail>
</OTA_AirDetailsRQ>
```

响应示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirDetailsRS>
  <FlightDetails TotalFlightTime="PT2H30M" TotalMiles="0" StopQuantity="0">
    <FlightLegDetails FlightNumber="1531" DepartureDateTime="2013-06-10T15:55:00"
ArrivalDateTime="2013-06-10T18:25:00">
      <DepartureAirport LocationCode="PEK"/>
      <ArrivalAirport LocationCode="SHA"/>
      <Equipment AirEquipType="330"/>
      <MarketingCabinAvailability CabinType="Y">
        <Meal MealCode="S"/>
      </MarketingCabinAvailability>
      <MarketingAirline Code="CA"/>
    </FlightLegDetails>
  </FlightDetails>
</OTA_AirDetailsRS>
```

## 7.2 以 PNR 号+航段序号方式进行查询

这种查询方式需要传入的参数包括：客户 office 号，PNR 号。  
程序会自动将该 PNR 与其中所有航段关联，查询出所需要的信息。  
调用示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirDetailsRQ>
  <POS>
    <Source PseudoCityCode="BJS191" />
  </POS>
  <BookingReferenceID ID="HR2D56"/>
</OTA_AirDetailsRQ>
```

响应示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirDetailsRS>
  <FlightDetails TotalFlightTime="PT1H40M" TotalMiles="0" StopQuantity="0">
    <FlightLegDetails FlightNumber="5641" DepartureDateTime="2013-07-05T18:15:00"
ArrivalDateTime="2013-07-05T19:55:00">
      <DepartureAirport LocationCode="XIY"/>
      <ArrivalAirport LocationCode="KWE"/>
      <Equipment AirEquipType="737"/>
      <MarketingCabinAvailability CabinType="Z">
        <Meal MealCode=" "/>
      </MarketingCabinAvailability>
      <MarketingAirline Code="KN"/>
    </FlightLegDetails>
  </FlightDetails>
</OTA_AirDetailsRS>
```

## 8.接口调用

航班详细信息查询服务提供基于 XML 文档规范的标准 HTTP 协议接口，通过在标准 HTTP 通道上传输 XML 格式的请求和数据来完成服务调用。服务的输入输出定义在标准的 XSD 文档中，用户根据自有系统的实现环境结合 XSD 文档进行数据到对象间的转换。在线分销提供服务接口 URL,用户使用 HTTP 协议按照服务定义正确构造请求输入，正确解析数据输出实现服务的调用。

## 8.1 接口申请

请与您所在城市的中国航信分支机构联系，资质审查通过后签署试用协议，即可获得试用接口。

## 8.2 接口认证

授权用户调用试用接口时需要使用已分配的账号和密码、并使用申请时的 IP。如果已变化 IP 请联系业务人员申请变更 IP。

## 8.3 接口地址

接口描述位置：<http://espeed.travelsky.com/develop/xml/AirDetails?xsd>

接口试用地址：<http://espeed.travelsky.com/develop/xml/AirDetails>

## 8.4 JAVA 调用示例

Java 调用 http 接口示例如下：

其中该样例程序只是为了展示如何接入使用

```
import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.zip.GZIPInputStream;
import java.util.zip.GZIPOutputStream;

import org.apache.commons.httpclient.Credentials;
import org.apache.commons.httpclient.DefaultHttpMethodRetryHandler;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpState;
import org.apache.commons.httpclient.HttpStatus;
import org.apache.commons.httpclient.UsernamePasswordCredentials;
import org.apache.commons.httpclient.auth.AuthScope;
import org.apache.commons.httpclient.methods.ByteArrayRequestEntity;
import org.apache.commons.httpclient.methods.PostMethod;
import org.apache.commons.httpclient.methods.RequestEntity;
import org.apache.commons.httpclient.params.HttpMethodParams;

/**
 * http 客户端调用示例(数据压缩版)
 */
```

```
public class HttpClientDemo {

    public static void main(String[] args) {

        //实际使用时，请设置正确的变量值
        //用户名
        String username = "用户名";
        //密码
        String pwd = "密码";
        //服务地址
        String serviceUrl = "服务地址";
        //请求 XML
        String reqStr = "test-xml";

        // 构造 HttpClient 的实例
        HttpClient httpClient = new HttpClient();
        //调用验证信息
        HttpState state = new HttpState();
        Credentials credentials = new UsernamePasswordCredentials(username, pwd);
        state.setCredentials(AuthScope.ANY, credentials);
        httpClient.setState(state);

        // 创建 POST 方法的实例
        PostMethod postMethod = new PostMethod(serviceUrl);
        // 使用系统提供的默认的恢复策略
        postMethod.getParams().setParameter(HttpMethodParams.RETRY_HANDLER,
            new DefaultHttpClientRetryHandler());
        try {
            // 请求参数的数据压缩
            ByteArrayOutputStream out = new ByteArrayOutputStream();
            GZIPOutputStream gzip = new GZIPOutputStream(out);
            if (reqStr != null && !"".equals(reqStr)) {
                gzip.write(reqStr.getBytes());
            }
            gzip.close();
            RequestEntity requestEntity = new ByteArrayRequestEntity(out
                .toByteArray());
            postMethod.setRequestEntity(requestEntity);
            postMethod.addRequestHeader("Content-Type",
                "text/html;charset=UTF-8");
            postMethod.addRequestHeader("accept-encoding", "gzip");
            postMethod.addRequestHeader("content-encoding", "gzip");

            // 执行 getMethod
```

```

        int statusCode = httpClient.executeMethod(postMethod);
        if (statusCode != HttpStatus.SC_OK) {
            throw new Exception("Invoke Get Method Failed, HttpStatus = "
                                + statusCode);
        }

        // 返回结果的数据解压
        InputStream is = postMethod.getResponseBodyAsStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(
            new GZIPInputStream(is)));

        StringBuffer respStr = new StringBuffer();
        String line = null;
        while ((line = br.readLine()) != null) {
            respStr.append(line);
        }

        // 打印结果
        System.out.println(respStr.toString());

        out.flush();
        out.close();
        is.close();
    } catch (Exception e) {
        // 发生致命的异常，可能是协议不对或者返回的内容有问题
        e.printStackTrace();
    } finally {
        // 释放连接
        postMethod.releaseConnection();
    }
}
}
}

```

## 8.5.NET 调用示例

.Net 调用 http 接口示例如下：

其中该样例程序只是为了展示如何接入使用

(1)建立一个 cs 文件,与服务器通信的类：

// 定义 xml 与服务器的通信

```
class OTA_XMLHTTP
```

```
{
```



```

string user;    //用户名
string pwd;    //密码
string url;    //服务地址
public OTA_XMLHTTP(string _user,string _pwd,string _url)
{
    user = _user;
    pwd = _pwd;
    url = _url;
}
// 获取服务器返回的字符串
public string GetResponse(string requestXml)
{
    string xmlString = "";    //要返回的 xml 字符串
    bool a= requestXml.Contains("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
    requestXml = requestXml.Replace("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
    "");
    try
    {
        //制定服务器地址
        HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(url);
        request.Method = WebRequestMethods.Http.Post; //设定 http 的传递方式
        request.ContentType = "application/x-www-form-urlencoded";
        //设定 http 的 header
        if (user != null && pwd != null)
        {
            string user_pwd = user + ":" + pwd;
            byte[] authBytes = Encoding.UTF8.GetBytes(user_pwd.ToCharArray());
            request.Headers.Add("Authorization", "Basic " +
            Convert.ToBase64String(authBytes));
            request.Headers.Add("Content-Encoding", "gzip");
            request.Headers.Add("Accept-Encoding", "gzip");
        }
        //将数据写入流中
        string para = "request";
        ASCIIEncoding encoding = new ASCIIEncoding();
        byte[] data = encoding.GetBytes(para + " = " + requestXml);
        request.ContentLength = data.Length;
        Stream stream = request.GetRequestStream();
        stream.Write(data, 0, data.Length);
        stream.Close();
        //建立获取 http 返回的 response
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();
        //读取 http 返回的字符串
        Stream responseStream = response.GetResponseStream();
    }
    catch { }
}

```

```

        if (response.ContentEncoding.ToLower().Contains("gzip"))
            responseStream = new GZipStream(responseStream,
CompressionMode.Decompress);
        else if (response.ContentEncoding.ToLower().Contains("deflate"))
            responseStream = new DeflateStream(responseStream,
CompressionMode.Decompress);
        StreamReader streamReader = new StreamReader(responseStream,
Encoding.UTF8);
        xmlString = streamReader.ReadToEnd();
        streamReader.Close();
    }
    catch (WebException e)
    {
        xmlString = e.Message;
    }
    return xmlString;
}
}

```

(2) 调用上面的文件

调用方法：

```

string user="用户名";
string pwd="密码";
string url="接口地址";
OTA_XMLHTTP xmlHttp = new OTA_XMLHTTP(user, pwd, url);
PonseXmlBox.Text = xmlHttp.GetResponse(requestXml);

```