

在线分销产品

电子客票挂起服务用户手册（非订单）

目录

1. 概述.....	1
2. 版本.....	1
3. 业务范围.....	2
4. 功能范围.....	2
5. 接口说明.....	3
5.1 接口结构.....	3
5.2 请求参数说明.....	4
5.3 响应参数说明.....	5
6. 错误说明.....	5
6.1 http 协议错误.....	5
6.2 接口错误.....	6
7. 参考样例.....	7
7.1 PNR+出票日期+Office 号.....	7
7.2 票号+Office 号.....	8
8. 接口调用.....	9
8.1 接口申请.....	9
8.2 接口认证.....	9
8.3 接口地址.....	9
8.4 JAVA 调用示例.....	10
8.5 .NET 调用示例.....	13

1. 概述

客票挂起服务是中国航信在线分销平台统一解决方案中专为在线分销代理人定制的电子客票产品，以便捷、准确的方式满足代理人在线更改电子客票挂起状态的需求。

2. 版本

主要内容	客票挂起				
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	2013/5/23	创建	赵高红		

3. 业务范围

- 挂起接口只对”open for use”状态的票号进行挂起。
- 用户传入票号、Office 号，对票号进行挂起操作。
- 用户传入 PNR 号、office 号、出票日期，进行挂起操作。
- office 号必须和出票 Office 一致。

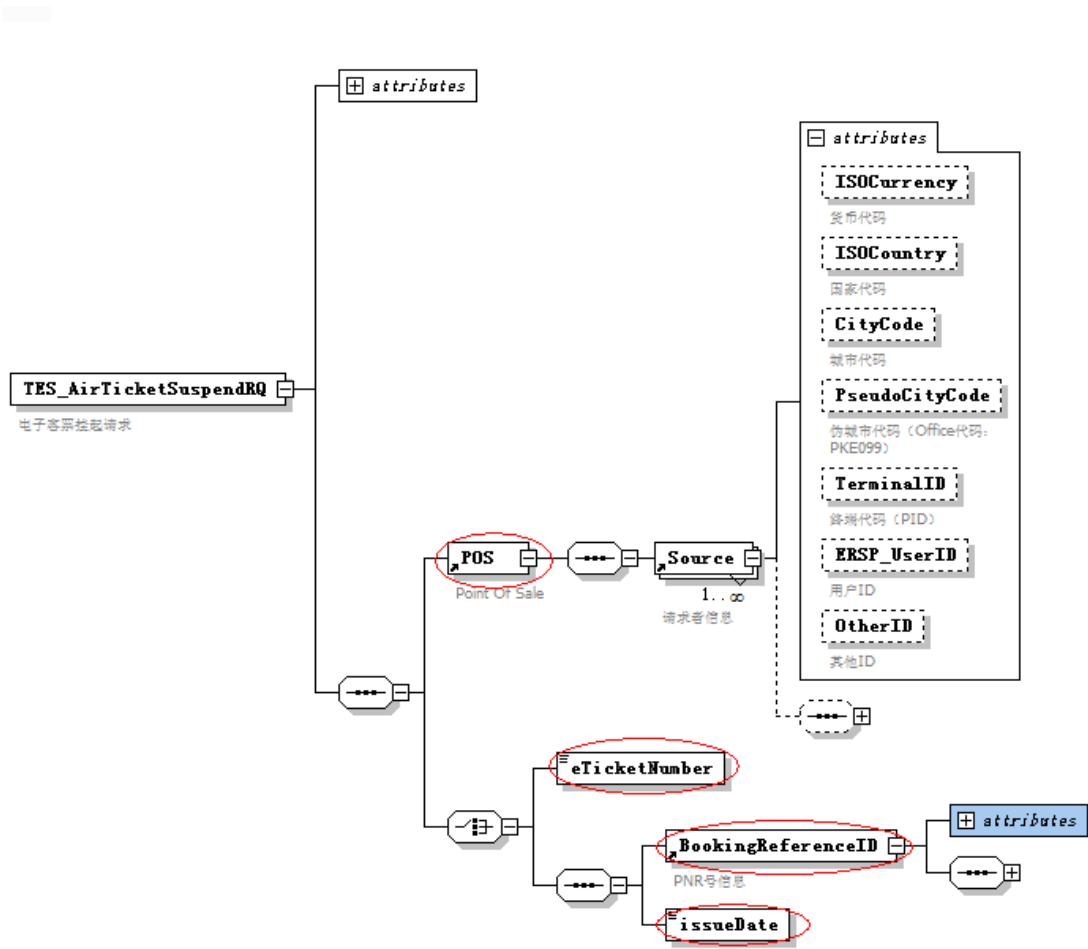
4. 功能范围

- 对 “OPEN FOR USE” 状态的票号进行挂起操作。
- 对 “OFFICE+单张票号” 进行挂起操作。
- 对 “OFFIC+PNR 号+出票日期” 进行挂起操作。

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：

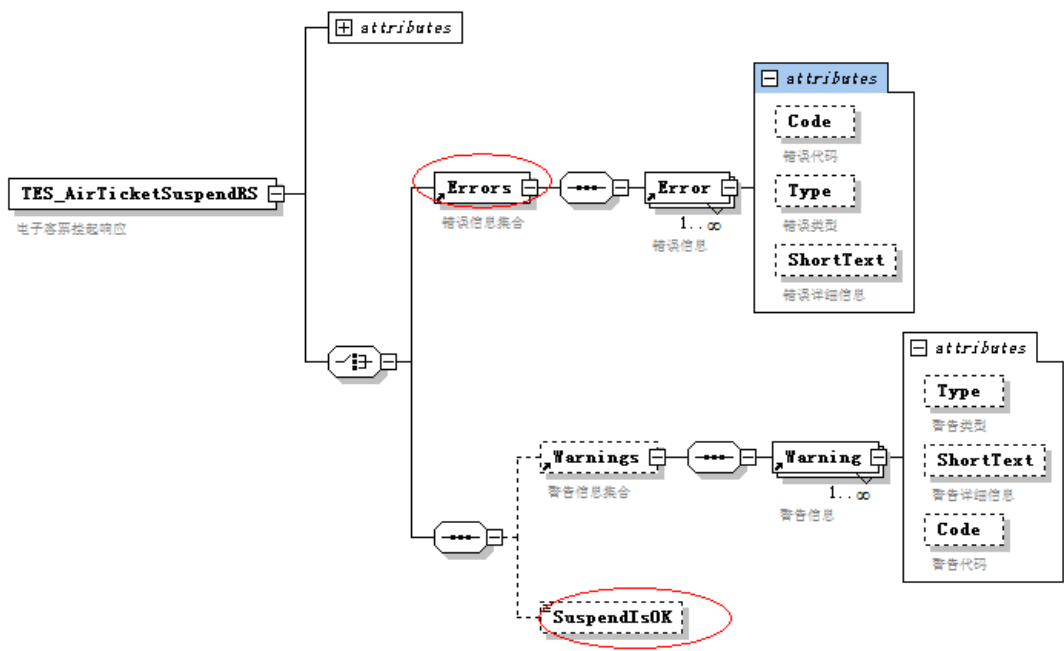
POS 节点：用于传递 PseudoCityCode 信息，即 Office 号。

eTicketNumber 节点：表示电子客票。

BookingReferenceID 节点：用来传递 PNR 信息。

issueDate 节点：表示出票日期。

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：用来记录挂起错误的票号信息。

SuspendIsOK 节点：表示挂起是否成功。

5.2 请求参数说明

元素名称	说明	Rep	类型	默认值	备注
TES_AirTicketSuspendRQ		1			
POS		1			
Source		1			
@PseudoCityCode		1	string		Office 号
eTicketNumber		0..1	string		电子票号
BookingReferenceID	PNR 号 信息	0..1			
@ID		0..1	string		PNR 号
@ID_Context		0..1	string		PNR 号上下文

issueDate	0..1	date	出票日期
-----------	------	------	------

5.3 响应参数说明

元素名称	说明	Rep	类型	默认值	备注
TES_AirfareAutoTicketRS		1			
Errors	错误信息	0..1			
Error		0..1			
@Code	错误代码	0..1			
@Type	错误类型	0..1			
@ShortText	错误详细信息	0..1			
Warnings		0..1			
Warning		0..1			
@Type	警告信息	0..1			
@ShortText	警告详细信息	0..1			
@Code	警告代码	0..1			
SuspendIsOK		0..1	string		挂起是否成功

6. 错误说明

6.1 http 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。详见下表：

系统错误

错误代码	错误描述	说明
-303		

业务错误

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
-500	office code is empty	office 代码为空
-560	pnr locator is empty	pnr 号为空
-710	date format is not valid	日期格式错误
-9999		未知错误
-1		服务异常
-2		输入错误
-100		系统不可用
-101		系统繁忙
-200		服务不可用
-201		服务繁忙
-202		用户无权限
-300		用户被锁定
-301		用户访问频率过高
-302		用户认证错误
-303		用户 office 号校验错误
-304		用户 office 号格式错误

-305		用户 IP 错误
-306		用户 IP 校验错误
-307		用户 IP 格式错误

7. 参考样例

7.1 PNR+出票日期+Office 号

下面以 票号：781-3952031493 为例子，进行客票挂起操作

输入样例

```
<?xml version="1.0" encoding="UTF-8"?>
<TES_AirTicketSuspendRQ>
  <POS>
    <Source PseudoCityCode="BJS187"></Source>
  </POS>
  <BookingReferenceID ID="JTGTSE" ></BookingReferenceID>
  <issueDate>2013-05-20 08:37:00</issueDate>
</TES_AirTicketSuspendRQ>
```

对应输出：

```
<TES_AirTicketSuspendRS>
  <SuspendIsOK>OK</SuspendIsOK>
</TES_AirTicketSuspendRS>
```

黑屏 RT 结果的输出：

```

▶DETR:TN/781-3952031493
ISSUED BY: CHINA EASTERN AIRLINES   ORG/DST: BJS/JGN           BSP-D
E/R: 不得签转
TOUR CODE:
PASSENGER: 刘克利
EXCH:
O FM:1PEK MU      2412  Y 10JUN 1120 OK YDM85           20K SUSPENDED
   T2-- RL:NJJGX1  /JGTSE1E
   TO: JGN
FC: 10JUN13PEK MU JGN1600.00CNY1600.00END**8911593
FARE:           CNY 1600.00|FOP:CASH
TAX:            CNY 50.00CN|OI:
TAX:            CNY110.00YQ|
TOTAL:          CNY 1760.00|TKTN: 781-3952031493

```

7.2 票号+Office 号

下面以 票号：999-3948887261 为例子，进行客票挂起操作

输入样例

```

<?xml version="1.0" encoding="UTF-8"?>
<TES_AirTicketSuspendRQ>
  <POS>
    <Source PseudoCityCode="bjs187"></Source>
  </POS>
  <eTicketNumber>999-3948887261</eTicketNumber>
</TES_AirTicketSuspendRQ>

```

对应输出：

```

<TES_AirTicketSuspendRS>
  <SuspendIsOK>OK</SuspendIsOK>
</TES_AirTicketSuspendRS>

```

黑屏 RT 结果的输出：

```

▶DETR:TN/999-3948887261
ISSUED BY: AIR CHINA                      ORG/DST: BJS/SHA                      BSP-D
E/R: 不得退改签
TOUR CODE:
PASSENGER: WANG/YA
EXCH:
CONJ TKT:
O FM:1PEK CA      1883  Y 13JUL 0800 OK Y      20K SUSPENDED
      T3T2 RL:ND1D44  /JQ6R0D1E
      TO: PVG
FC: 13JUL13PEK CA PVG1130.00CNY1130.00END
FARE:           CNY 1130.00|FOP:CASH
TAX:            CNY 50.00CN|OI:
TAX:            CNY140.00YQ|
TOTAL:          CNY 1320.00|TKTN: 999-3948887261

```

8. 接口调用

客票挂起服务提供基于 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/AirTicketSuspend?xsd>

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

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 DefaultHttpMethodRetryHandler());

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
    HttpResponseMessage response = (HttpResponseMessage)request.GetResponse();
    //读取 http 返回的字符串
    Stream responseStream = response.GetResponseStream();
    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);
```