

在线分销产品
国内退改签规则查询

目录

- 1. 概述..... 1
- 2. 版本..... 1
- 3. 业务范围..... 1
- 4. 功能范围..... 2
- 5. 接口说明..... 2
 - 5.1 接口结构..... 2
 - 5.2 请求参数说明..... 3
 - 5.3 响应参数说明..... 5
- 6. 错误说明..... 6
 - 6.1Http 协议错误..... 6
 - 6.2 接口错误..... 7
- 7. 参考样例..... 8
 - 7.1 查询实例..... 8
- 8. 接口调用..... 13
 - 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..... 18

1. 概述

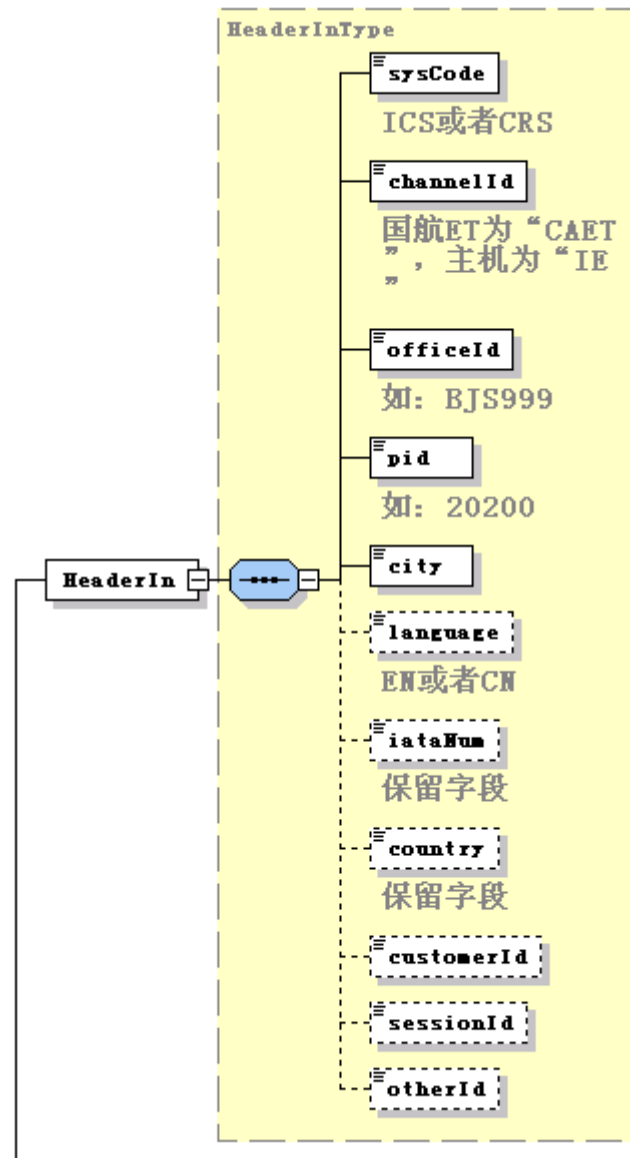
国内退改签规则查询服务(AirRuleDisplay)为用户提供国内退改签规则查询。

2. 版本

主要内容	国内退改签规则查询				
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	2013.05.27	创建	程名		

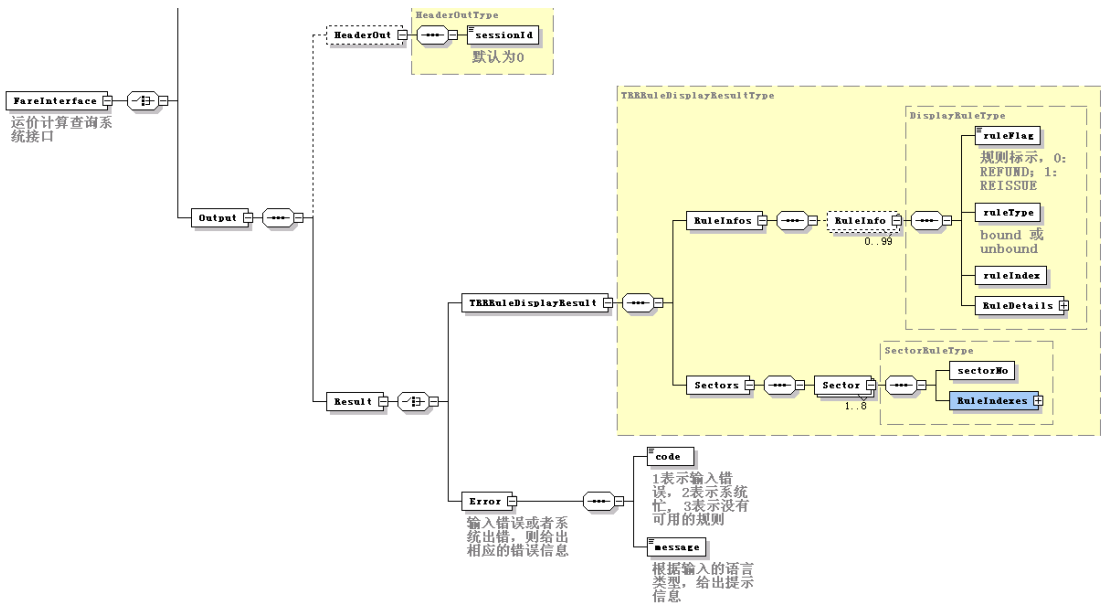
3. 业务范围

提供国内的退改签规则的查询服务



HeaderIn 节点

5.1.2 响应结构



响应结构

5.2 请求参数说明

元素名称	说明	Rep	类型	默认值	备注
Input		1			
HeaderIn		1			
Syscode	出票的系统代码	1			
ChannelId	系统类型，默认为 1E	1	string		国航 ET 为“CAET”,主机为“1E”
OfficeId	Office 号	1	string		
Pid	PID 号	1	string		
City	城市	1	string		
Language	语言，非必填	0..1	string		
IataNum	非必填，保留字段	0..1	string		

Country	非必填，保留字段	0..1	string	
CustomerId	非必填，保留字段	0..1	string	
SessionId	非必填，保留字段	0..1	string	
OtherId	非必填，保留字段	0..1	string	
Request		1		
TRRRuleDisplayRequest		1		
PricingDateTime		1	string	
Date	运价计算日期	1	string	比如 28JUL06
Time	运价计算时间	1	string	比如 1400
PassengerType	旅客类型	0..1	string	指令中输入特殊旅客类型（CH/IN/GM/JC/折扣代码等），如果没有输入旅客类型，则表示普通成人 例： CH
RequestRuleType	请求类型	1	int	请求类型，包括 1：绑定；2：非绑定；3：绑定及非绑定
Fn	票价组 Fn	1	string	比如： FCNY1130.00/SCNY1130.00/C3.00/XCNY150.00/TCNY50.00CN/TCNY100.00YQ
Fc	票价计算组 Fc	1	string	比如: PEK CA PVG 1130.00YCN1130.00END
Rmk		1		
Ot	Ot 项	1	string	

Fb	保留字段	1	string
Be	保留字段	1	string
Sectors	航段信息	1	
Sector		1..8	
SectorNo	航段序号	1	int
DeptDateTime	出发时间	1	string
Date	出发日期	1	string
Time	出发时间	1	string
Week	星期几出发	1	string
OriAirport	出发地点	1	string
DesAirport	到达地点	1	string
Carrier	承运公司代 码	1	string
BookingClass	订座舱位	1	string
FareBasis	运价基础	1	string
Ypub	Y 舱公布运 价		string

5.3 响应参数说明

下表是对响应输出中常用的属性字段进行说明，其他字段可参考 XSD。

元素名称	说明	Rep	类型	默认值	备注
Output		1			
Result	成功的标 志	0..1			如果成功包含此项
TRRRuleDisplayResult		1			

RuleInfos		1		
RuleInfo		0..99		
RuleFlag	规则标志	1	int	
RuleType	规则类型	1	int	
RuleIndex	规则索引	1	int	
RuleDetails		1		
RuleDetail		规则详情	0..99	
DetailNo	详情编号	1	int	
Text	详情内容	1	string	
Language	语言	1	string	
Sectors		1		
Sector		1..8		
SectorNo	航段编号	1	int	
RuleIndexes		规则索引 集合	1	
RuleIndex	规则索引	1..99	int	
Error		错误信息	0..1	
Code	错误代码	0..1	string	
Message	错误类型	0..1	string	对应错误类型名称

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
-303	inner service error	内部服务异常

业务错误

错误代码	错误描述	说明
-1	Biz Error	业务异常

7. 参考样例

输入请求的模板说明如下，其他信息可根据业务需要对应到相应的节点中，各个节点说明可参考 TES_AirRuleDisplay.xsd。

7.1 查询实例

请求示例：

```
<FareInterface>
  <Input>
    <HeaderIn>
      <sysCode>ICS</sysCode>
      <channelId>1E</channelId>
```

<officeId>PEK099</officeId>

<pid>11966</pid>

<city>BJS</city>

<language>CN</language>

</HeaderIn>

<Request>

<TRRRuleDisplayRequest>

<PricingDateTime>

<date>06DEC12</date>

<time>1752</time>

</PricingDateTime>

<passengerType/>

<requestRuleType>3</requestRuleType>

<fn>FCNY310.00/SCNY310.00/C3.00/TEXEMPTCN/TCNY --YQ</fn>

<fc>PEK CA CAN 310.00DIV1 CNY310.00END</fc>

<Rmk>

<ot>A/0/11966/3-1CA3911N1</ot>

<fb/>

<be/>

</Rmk>

<Sectors>

<Sector>

<sectorNo>1</sectorNo>

<DeptDateTime>

<date>06DEC12</date>

<time>1752</time>

</DeptDateTime>

<oriAirport>PEK</oriAirport>

<desAirport>CAN</desAirport>

```
        <carrier>CA</carrier>

        <bookingClass>B</bookingClass>

        <fareBasis>TRD_B</fareBasis>

        <ypub/>

    </Sector>

</Sectors>

</TRRRuleDisplayRequest>

</Request>

</Input>

</FareInterface>
```

响应示例:

```
<FareInterface>

  <Output>

    <Result>

      <TRRRuleDisplayResult>

        <RuleInfos>

          <RuleInfo>

            <ruleFlag>REFUND</ruleFlag>

            <ruleType>BOUND</ruleType>

            <ruleIndex>0</ruleIndex>

            <RuleDetails>

              <RuleDetail>

                <detailNo>0</detailNo>

                <text>hyz391refund_vol</text>

              </RuleDetail>

              <RuleDetail>

                <detailNo>1</detailNo>

                <text>hyz391refund_invol</text>

              </RuleDetail>

            </RuleDetails>

          </RuleInfo>

        </RuleInfos>

      </TRRRuleDisplayResult>

    </Result>

  </Output>

</FareInterface>
```

</RuleInfo>

<RuleInfo>

<ruleFlag>REISSUE</ruleFlag>

<ruleType>BOUND</ruleType>

<ruleIndex>1</ruleIndex>

<RuleDetails>

<RuleDetail>

<detailNo>0</detailNo>

<text>hyz391_reissue_vol</text>

</RuleDetail>

<RuleDetail>

<detailNo>1</detailNo>

<text>hyz391_reissue_invol</text>

</RuleDetail>

</RuleDetails>

</RuleInfo>

<RuleInfo>

<ruleFlag>REFUND</ruleFlag>

<ruleType>UNBOUND</ruleType>

<ruleIndex>2</ruleIndex>

<RuleDetails>

<RuleDetail>

<detailNo>0</detailNo>

<text>REFUND1EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>1</detailNo>

<text>REFUND1CN</text>

</RuleDetail>

<RuleDetail>

<detailNo>2</detailNo>

<text>REFUND2EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>3</detailNo>

<text>REFUND2CN</text>

</RuleDetail>

<RuleDetail>

<detailNo>4</detailNo>

<text>REFUND3EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>5</detailNo>

<text>REFUND3CN</text>

</RuleDetail>

</RuleDetails>

</RuleInfo>

<RuleInfo>

<ruleFlag>REISSUE</ruleFlag>

<ruleType>UNBOUND</ruleType>

<ruleIndex>3</ruleIndex>

<RuleDetails>

<RuleDetail>

<detailNo>0</detailNo>

<text>REISSUE1EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>1</detailNo>

<text>REISSUE1CN</text>

</RuleDetail>

<RuleDetail>

<detailNo>2</detailNo>

<text>REISSUE2EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>3</detailNo>

<text>REISSUE2CN</text>

</RuleDetail>

<RuleDetail>

<detailNo>4</detailNo>

<text>REISSUE3EN</text>

</RuleDetail>

<RuleDetail>

<detailNo>5</detailNo>

<text>REISSUE3CN</text>

</RuleDetail>

</RuleDetails>

</RuleInfo>

</RuleInfos>

<Sectors>

<Sector>

<sectorNo>1</sectorNo>

<RuleIndexes>

<ruleIndex>0</ruleIndex>

<ruleIndex>1</ruleIndex>

<ruleIndex>2</ruleIndex>

<ruleIndex>3</ruleIndex>

```
        </RuleIndexes>

        </Sector>

    </Sectors>

    </TRRRuleDisplayResult>

</Result>

</Output>

</FareInterface>
```

8. 接口调用

国内退改签规则查询服务提供基于 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/AirRuleDisplay/D?xsd>

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

8.4 JAVA 调用示例

Java 调用 http 接口示例如下：

其中该调用示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 = "请求 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：

其中该调用示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();

        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="服务地址";

```

```
OTA_XMLHTTP xmlHttp = new OTA_XMLHTTP(user, pwd, url);
```

```
PonseXmlBox.Text = xmlHttp.GetResponse(requestXml);
```