

在线分销产品
PNR 结构化信息提取服务用户手册

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
5.1.1 请求结构.....	2
5.1.2 响应结构.....	4
5.2 请求参数说明.....	4
5.3 响应参数说明.....	5
6. 错误说明.....	8
6.1 http 协议错误.....	8
6.2 接口错误.....	8
7. 参考样例.....	9
7.1 PNR 提取.....	9
8. 接口调用.....	11
8.1 接口申请.....	11
8.2 接口认证.....	11
8.3 接口地址.....	12
8.4 JAVA 调用示例.....	12
8.5 .NET 调用示例.....	14

1. 概述

PNR 结构化信息提取服务通过主机 RT 指令查询主机 CRS 系统中的 PNR 记录, 提供简单有效的实时查询通道, 提取指定 PNR 号的结构化记录信息。

2. 版本

主要内容	PNR 结构化信息提取				
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	20130523	创建	刁华林		

3. 业务范围

对应主机 RT 指令, 提取 PNR 记录信息。

4. 功能范围

- 对应主机 RT 指令，提取 PNR 记录信息
- 提供开放的结构化数据
- 数据实时准确

5. 接口说明

5.1 接口结构

5.1.1 请求结构

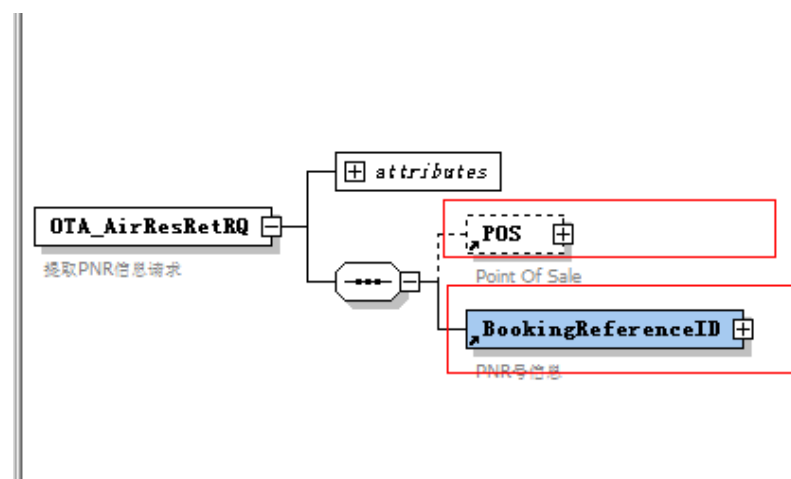


图 5.1 OTA_AirResRetRQ 节点结构

其中标识红色方框的节点结构如下：

POS 节点：

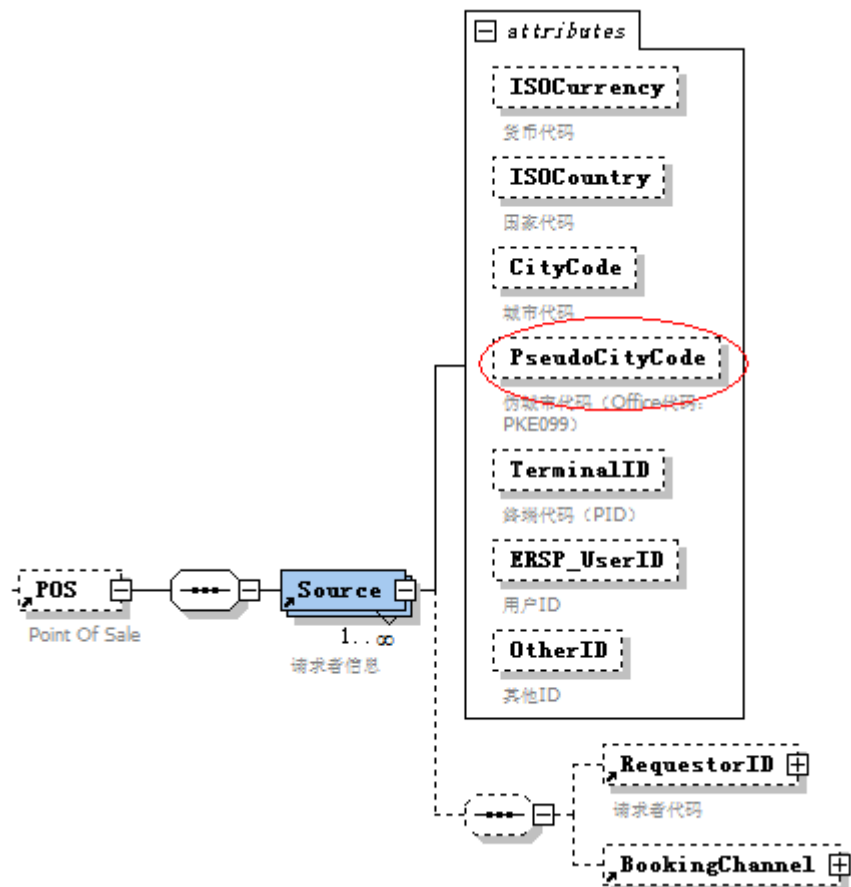


图 5.2 POS 节点结构

POS 节点只有一个，POS 节点下存放一个 Source 节点。Source 节点下 PseudoCityCode 节点存放代理人 Office。

BookingReferenceID 节点：

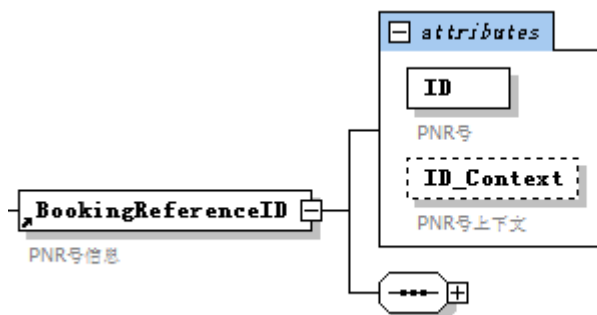


图 5.3 BookingReferenceID 节点结构

BookingReferenceID 节点只有一个，存放 PNR 信息。

5.1.2 响应结构

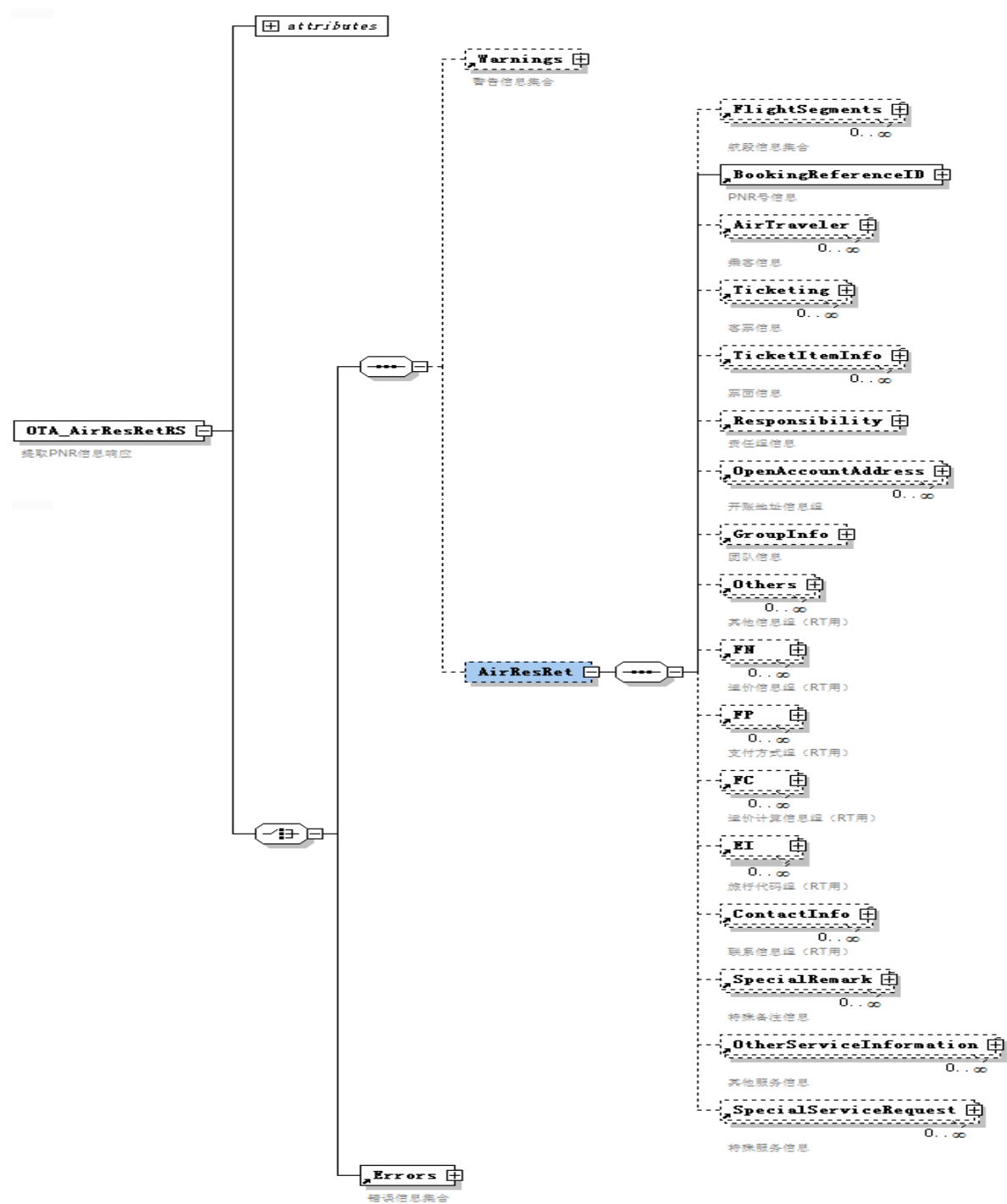


图 5.4 OTA_AirResRetRS 结构

5.2 请求参数说明

元素名称	说明	Rep	类型	默认	备注
OTA_AirResRetRQ		1			
POS		1			
Source		1			
@PseudoCityCode	代理人 Office 号	1	String		
BookingReferenceID		1			
@ID	PNR 号	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_AirResRetRS		1			
Errors	错误信息	0..1			
Error		0...n			
@Code	错误代码	0..1	String		
@Type	错误类型	0..1	String		
@ShortText	错误详细信息	0..1	String		
Warnings		0..1			
Warning		0..n			
@Type	警告信息	0..1	String		
@ShortText	警告详细信息	0..1	String		
@Code	警告代码	0..1	String		
AirResRet					
FlightSegments	航段信息集合	0...1			
FlightSegment	航段信息	0...n			
@FlightNumber	航班号	0...1	String		
@ArrivalDateTime	到达时间	0...1	String		
@NumberInParty	航段包含旅客数	0...1	String		
@DepartureDateTime	出发时间	0...1	String		
@SegmentType	航段类型	0...1	String		
@CodeshareInd	代码共享标识	0...1	String		

@Ticket	票证类型	0...1	String
	指定计划		String
@ScheduleValidEndDate	航班的有 效其实日 期	0...1	
@IsChanged	是否变更	0...1	String
@RPH	标识	0...1	String
@Status	航段行动 代码	0...1	String
BookingReferenceID		1	
@ID	PNR 号	0...1	String
AirTraveler		0...n	
	旅客信息		String
@PNR	在 PNR 中 的行号	0...1	
@PassengerTypeCode	旅客类型	0...1	String
PersonName	姓名信息	0...1	
@Surname	姓名的姓	0...1	String
@NamePNR	姓名 PNR	0...1	String
PassengerTypeQuantity	乘客类型 数量信息	0...1	String
@Age	年龄	0...1	String
Ticketing	客票信息	0...n	
@IsIssued	是否出票	0...1	String
@OfficeCode	出票 Office	0...1	String
@IssuedType	出票组类 型	0...1	String
@TicketTimeLimit	出票时限	0...1	String
	出票信息		String
@RPH	在 PNR 中 的行号	0...1	
@Remark	客票备注	0...1	String
TicketItemInfo	票面信息	0...n	
Responsibility	责任组信 息	1	
	读 PNR 在 CRS 系统 中对应的 PNR 编号		String
@PNRno		0...1	
@CRS	CRS 系统 标识	0...1	String
@OfficeCode	责任组	0...1	String

	Office 号		
@RPH	责任组信息在 PNR 中的行号	0...1	String
OpenAccountAddress	开账地址信息组	0...1	
GroupInfo	团队信息	1	
Others	其它信息组（RT）用	0...n	
FN	运价信息组	0...n	
FP	支付方式组(RT)用	0...n	
FC	运价计算信息组（RT）用	0...n	
EI	运价信息组（RT）用	0...n	
OI	OI 组（RT）用	0...n	
TC	旅行代码组（RT）用	0...n	
ContactInfo	联系信息组（RT）用	0...n	
@ContactCity	联系信息所在城市	0...1	String
@RPH	联系信息在 PNR 中的行号	0...1	String
@ContactInfo	联系信息内容	0...1	String
SpecialRemark	特殊备注信息	0...n	
@RemarkType	特殊备注类型	0...1	String
@RPH	特殊备注在 PNR 中的行号	0...1	String
@Text	备注信息	0...1	String
OtherServiceInformation	其它服务	0...n	

	信息		
@RPH	其它信息 服务在 PNR 中行 号	0...1	String
@Code		0...1	String
@Text	其它信息 服务内容	0...1	String
SpecialServiceRequest	特殊服务 信息	0...n	
@ServiceQuantity		0...1	String
@SSRCode	服务类型 代码	0...1	String
@BirthDate		0...1	String
@RPH	特殊服务 信息在 PNR 中行 号	0...1	String
@Text	特殊服务 信息内容	0...1	String

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，<Warnings>节点记录了警告信息。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!	内部服务错误

业务错误

错误代码	错误描述	中文说明
-1	Biz Error	业务异常
-400	format of xml string is not valid	xml 内容格式错误
-402	input error	输入错误
-10060	pnr does not exist	pnr 不存在
-10070	pnr has been cancelled already	pnr 已被取消
-9999	unknown error	未知错误

7. 参考样例

7.1PNR 提取

输入样例

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirResRetRQ xmlns="http://espeed.travelsky.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" EchoToken="String"
  TimeStamp="String" Version="String" Target="String">
  <POS>
    <Source PseudoCityCode="PEK099" />
  </POS>
  <BookingReferenceID ID="HQ8WEW">
    </BookingReferenceID>
  </OTA_AirResRetRQ>
```

对应输出：

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirResRetRS xmlns="http://espeed.travelsky.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" EchoToken="String"
  TimeStamp="String" Version="String" Target="String" SequenceNmbr="String">
  <AirResRet>
    <FlightSegments>
      <FlightSegment FlightNumber="1355" ArrivalDateTime="2011-08-11T20:20:00.0Z"
        NumberInParty="3" DepartureDateTime="2011-08-11T16:35:00.0Z"
        SegmentType="NORMAL" CodeshareInd="false" Ticket="eTicket">
```

```
ScheduleValidEndDate="1967-08-13" IsChanged="false" RPH="3" Status="RR">
  <DepartureAirport LocationCode="PEK" />
  <ArrivalAirport LocationCode="HAK" />
  <MarketingAirline CodeContext="String"
    CompanyShortName="String" Code="CA" />
  <BookingClassAvail ResBookDesigCode="A">
    <SubClass></SubClass>
  </BookingClassAvail>
</FlightSegment>
</FlightSegments>
<BookingReferenceID ID="HQ8WEW">
</BookingReferenceID>
<AirTraveler RPH="1" PassengerTypeCode="ADT">
  <PersonName>
    <Surname>郭应禄</Surname>
    <NamePNR>郭应禄</NamePNR>
  </PersonName>
  <PassengerTypeQuantity Age="0" />
</AirTraveler>
<AirTraveler RPH="2" PassengerTypeCode="ADT">
  <PersonName>
    <Surname>殷大奎</Surname>
    <NamePNR>殷大奎</NamePNR>
  </PersonName>
  <PassengerTypeQuantity Age="0" />
</AirTraveler>
<Ticketing IsIssued="false" OfficeCode="SHA320" IssuedType="TL"
  TicketTimeLimit="2011-08-10T18:00:00.0Z" RPH="0" Remark="TL/1800/10AUG/SHA320">
</Ticketing>

<Responsibility PNRno="" CRS="" OfficeCode="SHA666"
  RPH="17" />
<ContactInfo ContactCity="SHA" RPH="0"
  ContactInfo="SHA/T SHA/T 021-52918766/SHA GUANG FA AIR TICKET SERVICE
CO.,LTD/ZHOU JIAN
ABCDEFG">
  <TravelerRefNumber RPH="4" />
</ContactInfo>
<SpecialRemark RemarkType="" RPH="16">
  <Text>CA/NYF0G6</Text>

</SpecialRemark>
<OtherServiceInformation RPH="15" Code="">
  <Text>1E CAET TN/9992162073572-9992162073573</Text>
```

```
</OtherServiceInformation>
<SpecialServiceRequest ServiceQuantity="3"
    SSRCode="FOID" BirthDate="1967-08-13" RPH="7">
    <Text>FOID</Text>
</SpecialServiceRequest>
</AirResRet>
</OTA_AirResRetRS>
```

8. 接口调用

PNR 结构化信息提取服务提供基于 XML 文档规范的标准 HTTP 协议接口，通过在标准 HTTP 通道上传 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。在线分销平台提供服务接口 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/AirResRet?xsd>

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

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

```