

在线分销产品
PNR 原文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.....	2
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.....	5
5.3 响应参数说明.....	5
6. 错误说明.....	6
6.1 http 协议错误.....	6
6.2 接口错误.....	7
7. 参考样例.....	7
7.1 根据 PNR 状态和 PNR 号提取历史 PNR 信息——RT:C.....	7
7.2 根据 PNR 状态和 PNR 号提取现行 PNR 信息——RT.....	8
8. 接口调用.....	9
8.1 接口申请.....	9
8.2 接口认证.....	10
8.3 接口地址.....	10
8.4 JAVA 调用示例.....	10
8.5.NET 调用示例.....	13

1.概述

PNR 原文信息提取服务用于从主机中提取 PNR 信息。该服务提供了方便快捷的在线查询通道，以方便代理人客户在线查看 PNR 信息。

2.版本

主要内容	PNR 原文信息提取				
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	2013/05/23	创建	李权		

3.业务范围

查询 PNR 现行信息和 PNR 历史信息。

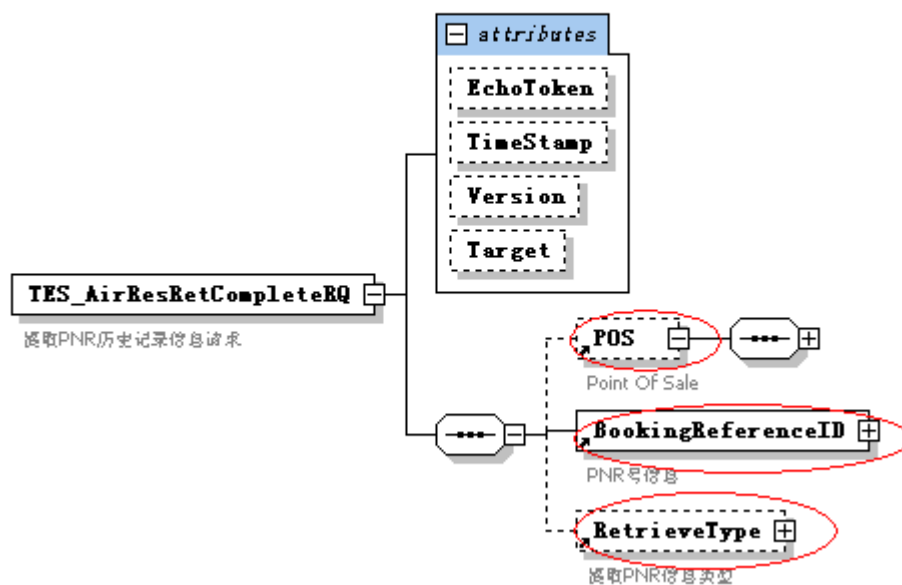
4.功能范围

- 提取 PNR 现行信息；
- 提取 PNR 历史信息；

5.接口说明

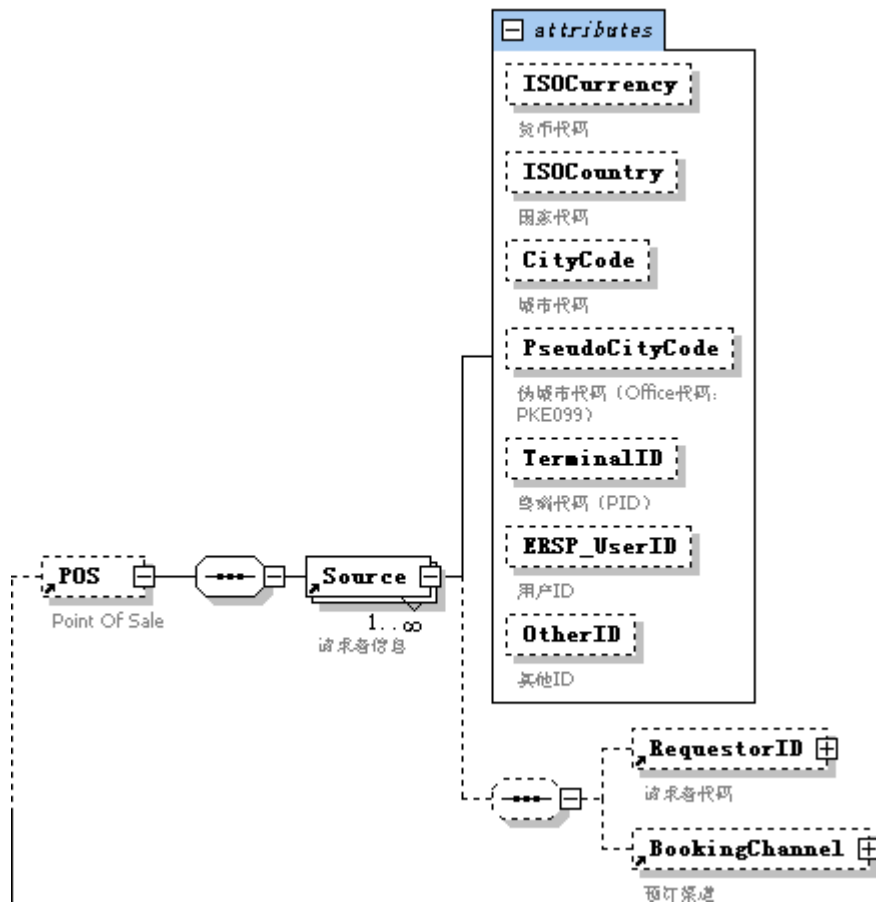
5.1 接口结构

5.1.1 请求结构



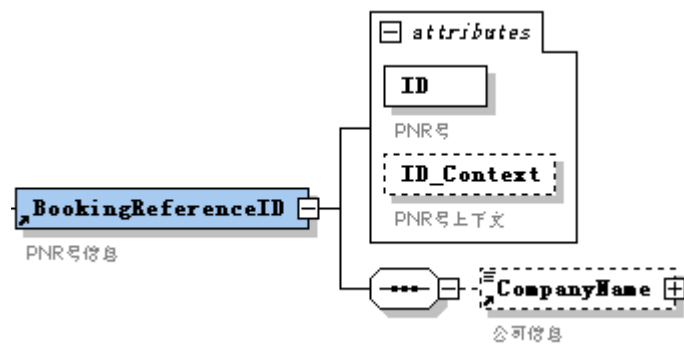
其中标识红色椭圆线的节点结构如下：

POS 节点



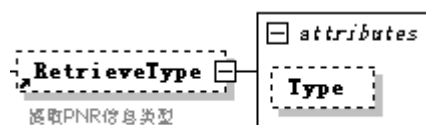
POS 节点存放的是零售商信息：如 office 号

BookingReferenceID 节点



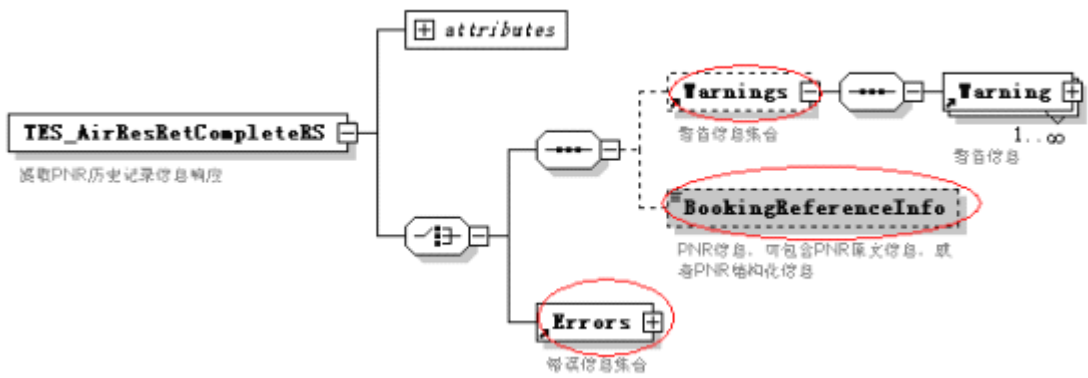
BookingReferenceID 节点存放 PNR 信息，其中 ID 存放 PNR 号，ID_Context 存放 PNR 上下文值可以为 PNR，ORDER，在此接口为 PNR

RetrieveType 节点

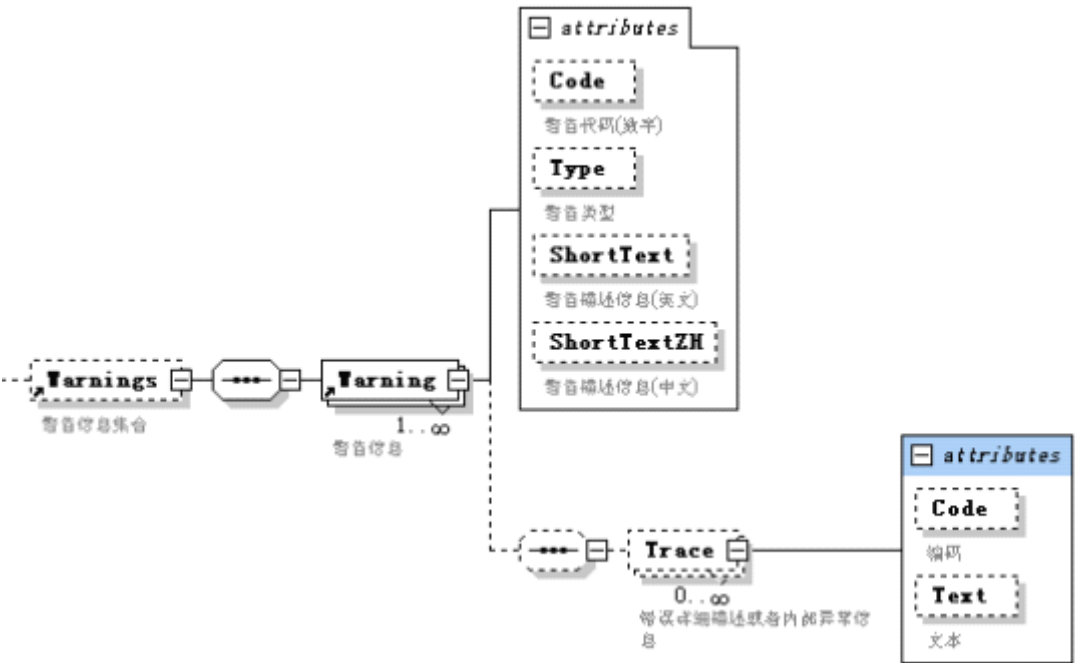


RetrieveType 节点存放 PNR 类型，值可以为：ACTIVE_RAW 或 HISTORY_RAW;

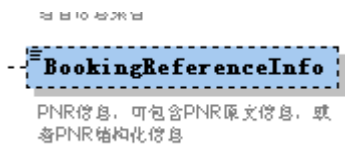
5.1.2 响应结构



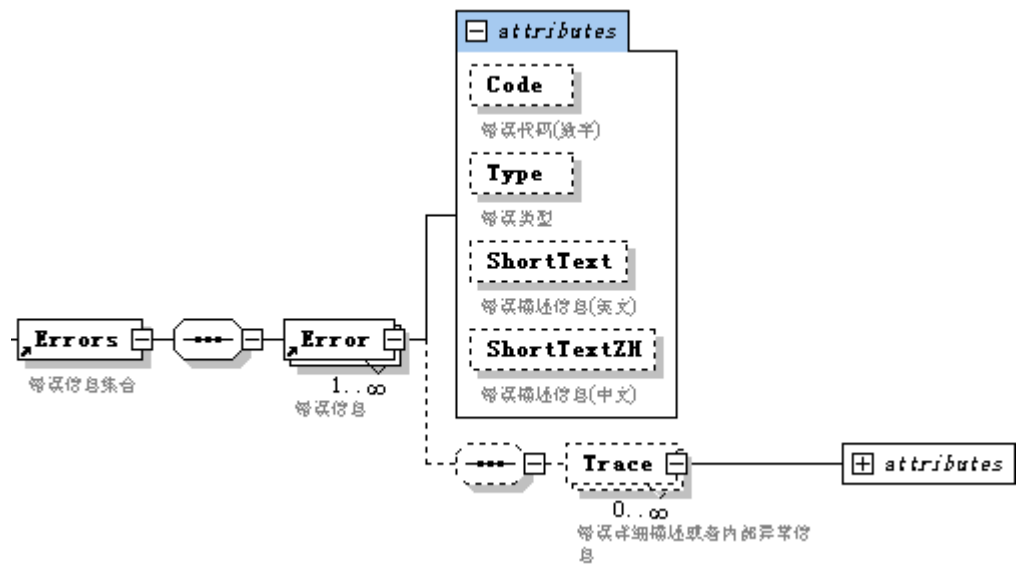
其中标识红色椭圆线的节点结构如下：
Warnings 节点



Warnings 节点存放返回信息的警告信息。
BookingReferenceInfo 节点



BookingReferenceInfo 节点存放查询返回的 PNR 信息；
Errors 节点



Errors 节点存放调用接口返回的错误信息。

5.2 请求参数说明

元素名称	说明	Rep	类型	默认值	备注
TES_AirResRetCompleteRQ	PNR 信息提取请求	1			
POS	零售商信息	0..1			
Source	请求商信息	1			
@PseudoCityCode	Office 号	0..1	string		
BookingReferenceID	PNR 信息	1			
@ID	PNR 号	1	string		
@ID_Context	PNR 上下文	0..1	string		可以选择值：ORDER，PNR 此处填值为：PNR
RetrieveType	PNR 类型	0..1			
@Type	PNR 类型	0..1		ACTIV E_RAW	ACTIVE_RAW: 现行信息 HISTORY_RAW: 历史信息

5.3 响应参数说明

元素名称	说明	Rep	类型	默认值	备注
TES_AirResRetCompleteRS	提取 PNR 信息响应	1			

BookingReferenceInfo	PNR 原文信息	0..1	string	该节点存放 PNR 原文信息
Warnings	警告信息集	0..1		
Warning	警告详细信息	1..n		
@Code	警告信息代码	0..1	string	
@Type	警告类型	0..1	string	
@ShortText	警告英文信息	0..1	string	
@ShortTextZH	警告中文信息	0..1	string	
Trace	警告附加信息	0..1		
@Code	警告代码	0..1	string	
@Text	警告附加信息	0..1	string	
Errors	错误信息集	0..1		
Error	错误信息	1..n		
@Code	错误信息代码	0..1	string	
@Type	错误类型	0..1	string	
@ShortText	错误英文信息	0..1	string	
@ShortTextZH	错误中文信息	0..1	string	
Trace	错误附加信息	0..1		
@Code	错误代码	0..1	string	
@Text	错误附加信息	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。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

业务错误

错误代码	错误描述	说明
-1	Biz Error	业务异常
-400	format of xml string is not valid	xml 内容格式错误
-700	office code is not valid	office 号格式错误
-760	format of pnr locator is not valid	pnr 号格式错误
-9999	unknown error	未知错误 (IBE 异常)
-10060	pnr does not exist	pnr 不存在
-43001	retrieve type is not valid	提取 PNR 类型错误
-43010	no authority to retrieve pnr	无权限提取 PNR

7.参考样例

7.1 根据 PNR 号提取历史 PNR 信息

提取 PNR 历史信息请求用例：

<pre><?xml version="1.0" encoding="UTF-8"?> <TES_AirResRetCompleteRQ> <POS> <Source PseudoCityCode="PEK099"></pre>
--

```

        </Source>
    </POS>
    <BookingReferenceID ID="JPFYE2">
        <CompanyName/>
    </BookingReferenceID>
    <RetrieveType Type="HistoryRaw"/>
</TES_AirResRetCompleteRQ>

```

提取 PNR 历史信息响应用例：

```

<TES_AirResRetCompleteRS>
  <BookingReferenceInfo>
    003      HDQCA 9983 0151 29FEB12 /1
            1.WANG/QIANG(001)  JDGM0C
    001  2.  HO1252 Y   FR09MAR12PEKSHA HK1   0630 0855           E T2T2
            NN(001)  DK(001)  HK(001)
    001  3.BJS/T BJS/T-65538922/CHINA AIR SERVICE COMPANY/HOU ZHU ABCDEFG
    001  4.010-57651857
    001  5.TL/1200/08MAR12/BJS191
    003  6.SSRADTK 1E BY BJS01MAR12/0951 OR CXL HO1252 Y09MAR
    002  7.RMK CA/NJ2R44
    001  8.BJS191
    001      BJS191 8888 0151 29FEB12
    002      HDQCA 9983 0151 29FEB12 /RLC1
    003      HDQCA 9983 0151 29FEB12 /1
  </BookingReferenceInfo>
</TES_AirResRetCompleteRS>

```

7.2 根据 PNR 号提取现行 PNR 信息

提取 PNR 现行信息请求用例：

```

<?xml version="1.0" encoding="UTF-8"?>
<TES_AirResRetCompleteRQ>
  <POS>
    <Source PseudoCityCode="PEK099">
    </Source>
  </POS>
  <BookingReferenceID ID="JPFYE2">
    <CompanyName/>
  </BookingReferenceID>
  <RetrieveType Type="ActiveRaw"/>
</TES_AirResRetCompleteRQ>

```

提取 PNR 现行信息响应示例：

```
<TES_AirResRetCompleteRS>
  <BookingReferenceInfo>
    1.WANG/QIANG JDGM0C
    2. HO1252 Y FR09MAR12PEKSHA HK1 0630 0855 E T2T2
    3.BJS/T BJS/T-65538922/CHINA AIR SERVICE COMPANY/HOU ZHU ABCDEFG
    4.010-57651857
    5.TL/1200/08MAR12/BJS191
    6.SSRADTK 1E BY BJS01MAR12/0951 OR CXL HO1252 Y09MAR
    7.RMK CA/NJ2R44
    8.BJS191
  </BookingReferenceInfo>
</TES_AirResRetCompleteRS>
```

8.接口调用

PNR 原文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/AirResRetComplete?xsd>

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

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();
}

}

}

        throw new Exception(
            "Invoke Post Method Failed, HttpStatus = " + statusCode);
    }

    byte[] responseBody = postMethod.getResponseBody(); // 读取内容
    response = new String(responseBody);
    System.out.println("response string =\n" + response); // 返回内容
} catch (Exception e) {
    e.printStackTrace();
} finally {
    postMethod.releaseConnection(); // 释放连接
}

return response;
}

}

```

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);

```