Technical documentation

API Sample Codes V1.0.6

Contents

1. Introduction

This technical documentation contains details for developers who wish to integrate our services into their own applications. It gives detailed explanation of how to use our API through HTTP protocol.

Accessing our API through HTTP protocol is one of the best, fastest and reliable ways.

To use this API, you need to have an account, you can register at (http://txtplaza.pshc.uk/member/showsignupForm). When you sign up, you can set your API password, it has to be different from your account password for additional security. You can also specify which IP address can access the API.

Our API can be contacted by either HTTP POST or GET, although both methods are available, we recommend using POST method for larger data transfer.

Please note that all parameters are case sensitive.

2. Getting started

In order to use our API, you need an account, after creating your account, start configuring the API to be able to access it. Please follow these steps to get started:

Step 1. Creating your account

- Go to http://txtplaza.pshc.uk/member/showsignupForm .
- Enter your information to complete the form.
- Read our Terms & conditions and be sure you accept them before creating your account.
- Click on “Register” button.

Step 2. Configuring the API

- When you access your account, please click on “API Access” from your account menu.
- In the form shown, please choose an API password which has to be different from your account password for additional security. You can set the allowed IP address which can connect to our API using your login details. If you don’t want to specify an IP address, please leave that field blank.
- Click on “Submit” button.

3. Connecting to the API

Please using the following URL to connect to our API:

http://api.txtplaza.pshc.uk

Our API accepts both HTTP POST and GET methods. Although both methods are available, we recommend using POST method for larger data transfer.

4. Sample codes

4.1 Account commands

4.1.1 PHP

This code uses cURL (http://curl.haxx.se).

```
<?
$url = "http://api.txtplaza.pshc.uk";
$data = '<?xml version="1.0" standalone="no" ?>
       <request>
              <authentication>
                <username>USERNAME</username>
                <password>PASSWORD</password>
              </authentication>
              <action>memberLogin</action>
     </request>';

$ch = curl_init( $url );
curl_setopt( $ch, CURLOPT_POST, 1);
curl_setopt( $ch, CURLOPT_POSTFIELDS, $data);
curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt( $ch, CURLOPT_HEADER, 0);
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1);

$response = curl_exec( $ch );

echo $response;
?>
```

4.1.2 ASP

This code uses Msxml2.ServerXMLHTTP.3.0 library (http://msdn.microsoft.com/en-us/library/ms766431(VS.85).aspx) and MSXML.DOMDocument library (http://msdn.microsoft.com/en-us/library/aa468547.aspx).

```
<%@ Language=VBScript CodePage=65001 %>
<%option explicit%>
<%
function DoApiCall(sXml)
      dim XmlHttp
      dim PostURL
      dim ResponseXml
      PostURL="http://api.txtplaza.pshc.uk"

      Set XmlHttp =
Server.CreateObject("Msxml2.ServerXMLHTTP.3.0")
      XmlHttp.Open "POST", PostUrl, false

      XmlHttp.SetRequestHeader "Content-Type",
"application/xml; charset=UTF-8"
      XmlHttp.SetRequestHeader "Accept", "application/xml;
charset=UTF-8"
      XmlHttp.Send sXml
      ResponseXml = XmlHttp.ResponseText

      DoApiCall=ResponseXml
end function

dim sAPICALLXML
dim sAPIRESULTXML
dim oXML
sAPICALLXML="<?xml version=""1.0"" standalone=""no"" ?>" &_
       "<request>" &_
              "<authentication>" &_
                     "<username>USERNAME</username>" &_
                     "<password>PASSWORD</password>" &_
              "</authentication>" &_
           "<action>memberLogin</action>" &_
      "</request>"
```

```
sAPIRESULTXML=DoApiCall(sAPICALLXML)
response.write sAPIRESULTXML
Set oXML = Server.CreateObject("MSXML.DOMDocument")
oXML.Async = false
oXML.loadXml( sAPIRESULTXML )
If (oXML.parseError.errorCode <> 0) Then
     'ERROR loading XML
else
     'oXML object has XML result parsed
end if

%>
```

4.1.3 Java

```
import java.net.URL;
import java.net.URLConnection;
import java.net.HttpURLConnection;
import java.io.*;

public class API {
    private static final String API_URL =
"http://api.txtplaza.pshc.uk";
    private static final String XML = "<?xml version=\"1.0\"?>" +
                            "<request>" +
                            "<authentication>" +
                                 "<username>USERNAME</username>" +
                                 "<password>PASSWORD</password>" +
                            "</authentication>" +
                            "<action>memberLogin</action>" +
                       "</request>";

    public static void main(String[] args) {
        String xml = doApiCall();
        System.out.println(xml);
    }
```

```
private static String doApiCall() {
        String result = null;
        try {
            URL url = new URL(API_URL);
            URLConnection uc = url.openConnection();
            HttpURLConnection connection = (HttpURLConnection)
uc;
            connection.setDoOutput(true);
            connection.setDoInput(true);
            connection.setRequestMethod("POST");
            connection.addRequestProperty("Content-Type",
"application/xml; charset=UTF-8");
            connection.addRequestProperty("Accept",
"application/xml; charset=UTF-8");
            OutputStream out = connection.getOutputStream();
            Writer writer = new BufferedWriter(new
OutputStreamWriter(out, "UTF-8"));
            writer.write(XML);
            writer.flush();
            writer.close();
            InputStream in = connection.getInputStream();
            BufferedReader reader = new BufferedReader(new
InputStreamReader(in, "UTF-8"));
            StringBuffer sb = new StringBuffer();

            String line;
            while ((line = reader.readLine()) != null)
sb.append(line);
            reader.close();
            connection.disconnect();
            result = sb.toString();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return result;
    }
}
```

4.1.4 C# .NET

```
using System;
using System.Xml;
using System.Net;
using System.IO;
using System.Security.Cryptography;

Public static string GetSignature(String email, String
passphrase) {
      var enc = Encoding.ASCII;
      HMACSHA1 hmac = new HMACSHA1(enc.GetBytes(passphrase));
      hmac.Initialize();
      byte[] buffer = enc.GetBytes(email);
      return
BitConverter.ToString(hmac.ComputeHash(buffer)).Replace("-",
"").ToLower();
}

public static XmlDocument doApiCall(String sApiXML)
{
      //Declare XMLResponse document
      XmlDocument XMLResponse = null;

      //Declare an HTTP-specific implementation of the
WebRequest class.
      HttpWebRequest objHttpWebRequest;

      //Declare an HTTP-specific implementation of the
WebResponse class
      HttpWebResponse objHttpWebResponse = null;

      //Declare a generic view of a sequence of bytes
      Stream objRequestStream = null;
      Stream objResponseStream = null;

      //Declare XMLReader
      XmlTextReader objXMLReader;

      //Creates an HttpWebRequest for the API URL.
      objHttpWebRequest =
(HttpWebRequest)WebRequest.Create("http://api.txtplaza.pshc.uk"
);
```

```
      try
      {
            //Prepare HttpWebRequest properties
            byte[] bytes;
            bytes =
System.Text.Encoding.ASCII.GetBytes(sApiXML);
            objHttpWebRequest.Method = "POST";
            objHttpWebRequest.ContentLength = bytes.Length;
            objHttpWebRequest.ContentType = "application/xml;
encoding='utf-8'";
            objHttpWebRequest.Accept="application/xml;
encoding='utf-8'";

            //Get Stream object
            objRequestStream =
objHttpWebRequest.GetRequestStream();

            //Writes a sequence of bytes to the current stream
            objRequestStream.Write(bytes, 0, bytes.Length);

            //Close stream
            objRequestStream.Close();
            //Send and wait for a response.
            objHttpWebResponse =
(HttpWebResponse)objHttpWebRequest.GetResponse();

            if (objHttpWebResponse.StatusCode ==
HttpStatusCode.OK)
            {
                  //Get response stream
                  objResponseStream =
objHttpWebResponse.GetResponseStream();

                  //Load response stream into XMLReader
                  objXMLReader = new
XmlTextReader(objResponseStream);

                  //Declare XMLDocument
                  XmlDocument xmldoc = new XmlDocument();
                  xmldoc.Load(objXMLReader);

                  //Set XMLResponse object returned from
XMLReader
                  XMLResponse = xmldoc;

                  //Close XMLReader
                  objXMLReader.Close();
```

```
            }
            [illegible]
            [illegible]
      }
      catch (WebException we)
      {
            //TODO: Add exception handling
            throw new Exception(we.Message);
      }
      catch (Exception ex)
      {
            throw new Exception(ex.Message);
      }
      finally
      {
            //Close connections
            objRequestStream.Close();
            objResponseStream.Close();
            objHttpWebResponse.Close();
      }

      return XMLResponse;
}


String sApiXml= @"<?xml version=""1.0"" standalone=""no"" ?>
      <request>
            <authentication>
                  <username>USERNAME</username>
                  <password>PASSWORD</password>
            </authentication>
            <action>memberLogin</action>
      </request>";

XmlDocument ApiXMLResponse = null;
ApiXMLResponse=doApiCall(sApiXml);
System.Console.WriteLine(ApiXMLResponse.InnerXml);
```