

C# 调用Smartbi服务器提供的方法示例

使用C#通过 HTTP JSON的方式，调用Smartbi服务器提供的所有方法。

JSON在C#中已有很多实现，具体可参见<http://www.json.org/>。

本示例使用的是Json.NET(安装方法)，如下图：

- C#:
 - fastJSON.
 - JSON_checker.
 - Javrock.
 - **Json.NET - LINQ to JSON.**
 - LitJSON.
 - JSON for .NET.
 - JsonFx.
 - JSON@CodeTitans
 - How do I write my own parser?
 - JSONSharp.
 - JsonExSerializer.
 - fluent-json
 - Manatee Json
 - FastJsonParser

1、核心示例代码：

Form1.cs

```
using System;
using System.Net;
using System.Windows.Forms;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System.IO;
namespace HttpRequestTest
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            CookieContainer cookieContainer = new CookieContainer();
            String serviceUrl = "http://localhost:8060/smartbi";
            Object ret = RemoteInvoker.InvokeMethod(cookieContainer, serviceUrl, "UserService", "login", new
String[] { "admin", "manager" });
            if (ret.ToString().Equals("True"))
            {
                MessageBox.Show("");
            }
            else
            {
                MessageBox.Show("");
                return;
            }
            // ID("\\SQL\SQL")
            String bizViewId = "I2c94907127855da201278df551b62602";
            // ID
            String paramId = "OutputParameter.I2c94907127855da201278df551b62602._";
            //
            String paramValue = ",,,";
        }
    }
}
```

```

//
JSONObject pobj = new JSONObject();
pobj["id"] = paramId;
pobj["value"] = paramValue;
JSONArray arr = new JSONArray();
arr.Add(pobj);
//
String paramsJsonArrStr = JsonConvert.SerializeObject(arr);
//
int rowsPerPage = 2;
//
Boolean getTotalRows = true;
JSONObject obj = (JSONObject)RemoteInvoker.invokeMethod(cookieContainer, serviceUrl,
"BusinessViewService",
    "openLoadDataView", new Object[] { bizViewId, paramsJsonArrStr, rowsPerPage, getTotalRows });
String loadDataClientId = (String)obj["clientId"];
int totalRows = (int)obj["totalRowCount"];
JSONArray fieldNames = (JSONArray)obj["fieldNames"];
// JSONArray fieldTypes = (JSONArray)ret["fieldTypes"];
String fName = "";
for (int i = 0; i < fieldNames.Count; i++)
{
    fName += fieldNames[i].ToString() + "\t";
}
textBox1.AppendText(fName + "\n");
int pages = totalRows / rowsPerPage + (totalRows % rowsPerPage == 0 ? 0 : 1);
for (int pageNum = 0; pageNum < pages; pageNum++)
{
    JSONArray pageData = (JSONArray)RemoteInvoker.invokeMethod(cookieContainer, serviceUrl,
"BusinessViewService",
    "loadViewData", new Object[] { loadDataClientId, pageNum });
    for (int j = 0; j < pageData.Count; j++)
    {
        String line = "";
        JSONArray rowData = (JSONArray)pageData[j];
        for (int k = 0; k < rowData.Count; k++)
        {
            line += (String)rowData[k] + "\t";
        }
        textBox1.AppendText(line + "\n");
    }
}
RemoteInvoker.invokeMethod(cookieContainer, serviceUrl, "BusinessViewService",
"closeLoadDataView", new Object[] { loadDataClientId });
}
private void button2_Click(object sender, EventArgs e)
{
    CookieContainer cookieContainer = new CookieContainer();
    String serviceUrl = "http://localhost:8060/smartbi";
    Object ret = RemoteInvoker.invokeMethod(cookieContainer, serviceUrl, "UserService", "login", new
String[] { "admin", "manager" });
    if (ret.ToString().Equals("True"))
    {
        MessageBox.Show("");
    }
    else
    {
        MessageBox.Show("");
        return;
    }
    String resid = "I2c94908d162343520116234cb928003d";
    JSONObject reportRet = (JSONObject)RemoteInvoker.invokeMethod(cookieContainer, serviceUrl,
"ClientReportService", "openQuery", new String[] { resid });
    JSONObject reportBean = (JSONObject)reportRet["reportBean"];
    String clientId = (String)reportRet["clientId"];
    String downloadURL = serviceUrl + "/vision/ExportServlet?delimiter=,&type=EXCEL2007&clientId=" +
clientId;
    FileStream stream = new FileStream(@"E:\11.xlsx", FileMode.OpenOrCreate, FileAccess.Write);
    RemoteInvoker.InvokeDownload(cookieContainer, downloadURL, "post", stream);
    stream.Close();
    RemoteInvoker.invokeMethod(cookieContainer, serviceUrl, "ClientReportService",

```

```

"removeFromSession", new String[] { clientId });
    }
}

```

RemoteInvoker.cs

```

using System;
using System.IO;
using System.Net;
using System.Text;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System.Net.Cache;
namespace HttpRequestTest
{
    public class RemoteInvoker
    {
        public static String encode = "UTF-8";
        public static void setEncode(String settingEncode)
        {
            encode = settingEncode;
        }
        public static Object invokeMethod(CookieContainer cookieContainer,
            String serviceUrl, String serviceName,
            String methodName, Object[] methodParams)
        {
            String paramInfo = JsonConvert.SerializeObject(methodParams);
            return RemoteInvoker.invokeBIOfficeRMIService(cookieContainer, serviceUrl, serviceName,
methodName, paramInfo);
        }
        public static Object invokeBIOfficeRMIService(CookieContainer cookieContainer,
            String serviceUrl, String serviceName,
            String methodName, String paramInfo)
        {
            WebResponse response = null;
            HttpWebRequest req = null;
            try
            {
                byte[] bytes = Encoding.GetEncoding(encode).GetBytes(paramInfo);
                String temp = Encoding.Default.GetString(bytes);
                String requestURL = serviceUrl + "/vision/RMIServlet";
                String postData = "className=" + serviceName + "&methodName=" + methodName
                    + "&params=" + paramInfo;
                byte[] data = Encoding.GetEncoding(encode).GetBytes(postData);
                req = (HttpWebRequest)HttpWebRequest.Create(requestURL);
                req.CookieContainer = cookieContainer;
                req.Method = "POST";
                req.ContentType = "application/x-www-form-urlencoded";
                req.ContentLength = data.Length;
                Stream reqStream = req.GetRequestStream();
                reqStream.Write(data, 0, data.Length);
                reqStream.Close();
                response = req.GetResponse();
                Stream stream = response.GetResponseStream();
                StreamReader sr = new StreamReader(stream, Encoding.GetEncoding(encode));
                String result = sr.ReadToEnd();
                JObject jsonObj = JObject.Parse(result);
                if ((int)jsonObj["retCode"] == 0)
                {
                    return jsonObj["result"];
                }
            }
            catch
            {
            }
            finally
            {
            }
        }
    }
}

```

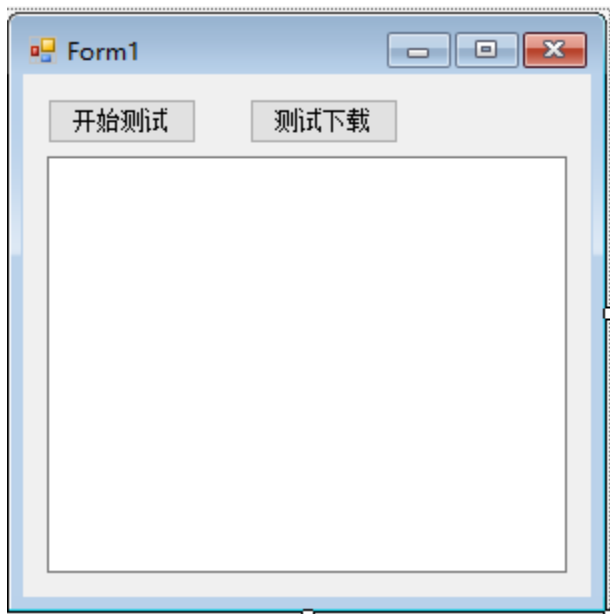
```

        response.Close();
        req.Abort();
    }
    return null;
}
public static void InvokeDownload(CookieContainer cookieContainer,
    String requestUrl, String method, byte[] data, Stream stream)
{
    WebResponse wr = null;
    try
    {
        var req = (HttpWebRequest)HttpWebRequest.Create(requestUrl);
        /* */
        req.CachePolicy = new HttpRequestCachePolicy(HttpRequestCacheLevel.NoCacheNoStore);
        req.Method = method;
        req.CookieContainer = cookieContainer;
        if (data != null)
        {
            req.ContentType = "application/x-www-form-urlencoded";
            Stream s = req.GetRequestStream();
            s.Write(data, 0, data.Length);
            s.Close();
        }
        wr = req.GetResponse();
        CopyStream(wr.GetResponseStream(), stream);
        req.Abort();
    }
    finally
    {
        try
        {
            wr.Close();
        }
        catch
        {
        }
    }
}

public static void CopyStream(Stream input, Stream output)
{
    byte[] buffer = new byte[32768];
    while (true)
    {
        int read = input.Read(buffer, 0, buffer.Length);
        if (read <= 0)
            return;
        output.Write(buffer, 0, read);
    }
}
}
}
}

```

2、运行效果:



3、示例工程(C#)源码:

[HttpRequestTest.rar](#)