C# JSON格式数据解析

1. C#对JSON解析需要的环境

在C#编写解析JSON时 需要下载json的插件包 Newtonsoft.Json.dll

使用开源的类库Newtonsoft.Json(下载地址http://json.codeplex.com/)。下载后加入工程就能用。通常可以使用JObject, JsonReader, JsonWriter处理。这种方式最通用，也最灵活，可以随时修改不爽的地方。

(1)使用JsonReader读Json字符串：

[csharp] view plaincopy

string jsonText =@"{""input"" : ""value"",""output"" : ""result""}";

JsonReader reader = new JsonTextReader(newStringReader(jsonText));

while (reader.Read())

{

Console.WriteLine(reader.TokenType + "\t\t" + reader.ValueType+ "\t\t" + reader.Value);

}

(2)使用JsonWriter写字符串：

[csharp] view plaincopy

StringWriter sw = new StringWriter();

JsonWriter writer = new JsonTextWriter(sw);

writer.WriteStartObject();

writer.WritePropertyName("input");

writer.WriteValue("value");

writer.WritePropertyName("output");

writer.WriteValue("result");

writer.WriteEndObject();

writer.Flush();

string jsonText =sw.GetStringBuilder().ToString();

Console.WriteLine(jsonText);

(3)使用JObject读写字符串：

[csharp] view plaincopy

JObject jo = JObject.Parse(jsonText);

string[] values =jo.Properties().Select(item => item.Value.ToString()).ToArray();

(4)使用JsonSerializer读写对象(基于JsonWriter与JsonReader):

数组型数据

[csharp] view plaincopy

string jsonArrayText1 ="[{'a':'a1','b':'b1'},{'a':'a2','b':'b2'}]";

JArray ja =(JArray)JsonConvert.DeserializeObject(jsonArrayText1);

string ja1a =ja[1]["a"].ToString();

//或者

JObject o = (JObject)ja[1];

string oa = o["a"].ToString();

1. JSON格式数据生成C#类

快速生成将JSON数据生成C#类，我们可以在网上找一下JSON生成C#类

例如：http://www.sojson.com/simple\_json.html

1. 下面模拟试衣镜接口环境做了对JSON数据的解析获取衣服的EPC值

我们将以下json字段进行解析：

{ "code": 100, "msg": "处理成功！", "extend": { "Dressing": { "EpcList": [ "000000000000000000000775", "000000000000000000000006" ], "recordNum": [ { "recordNum": 1, "epc": "000000000000000000000006" }, { "recordNum": 1, "epc": "000000000000000000000775" } ], "dressingInfo": [ { "startTime": "2017-07-17 13:38:29", "operatorType": 1, "epc": "000000000000000000000775", "endTime": "2017-07-17 13:38:34", "vipCardNo": "B000000000060D01DD4EFCDR" }, { "startTime": "2017-07-17 14:16:57", "operatorType": 1, "epc": "000000000000000000000006", "endTime": "2017-07-17 14:17:01", "vipCardNo": "B000000000060D01DD4EFCDR" } ], "VipList": [ "B000000000060D01DD4EFCDR" ] } } }

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Xml;

using Newtonsoft.Json; //导入引用Json

using System.Web;

using System.Net;

namespace jonsglxy

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://192.168.1.121:8099/"); //请求地址 应用过程中地址为接口对应地址如http://localhost:8080/RFIDSYS/mojing/call.do

request.Method = "GET"; //请求类型

request.ContentType = "text/html;charset=GB232";

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream myResponseStream = response.GetResponseStream();

StreamReader myStreamReader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string retString = myStreamReader.ReadToEnd(); //返回字符串

RootObject ss = JsonConvert.DeserializeObject<RootObject>(retString);

//RootObject jo = (RootObject)JsonConvert.DeserializeObject(retString);

string ll = "";

foreach (var item in ss.extend.Dressing.EpcList) //遍历Epclist中的值

{

ll += item;

}

textBox2.Text = ll;// 显示出取到的epclist中的值

textBox1.Text = retString; //显示获取的到的字符串

}

}

//以下内容是通过JSON转C#类的代码

public class RecordNum // RecordNum类

{

public string recordNum { get; set; }

public string epc { get; set; }

}

public class DressingInfo //DressingInfo类

{

public string startTime { get; set; }

public string operatorType { get; set; }

public string epc { get; set; }

public string endTime { get; set; }

public string vipCardNo { get; set; }

}

public class Dressing //dressing类

{

public List<string> EpcList { get; set; }

public List<RecordNum> recordNum { get; set; }

public List<DressingInfo> dressingInfo { get; set; }

public List<string> VipList { get; set; }

}

public class Extend //Extend类

{

public Dressing Dressing { get; set; }

}

public class RootObject //RootObject类

{

public string code { get; set; }

public string msg { get; set; }

public Extend extend { get; set; }

}

}