**Unity3D教程：利用UnityHttpClient类实现Http通信**

Posted on 2013年02月04日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 267 次

利用Warensoft.Unity.Communication.Client.UnityHttpClient类可以实现Http Get请求以及POST请求,并可以使用简单的获取本次请求所对对应的响应。可以使用该类完全代替内置的WWW类。

推荐使用UnityHttpClient替代WWW的原因有以下几点：

1>WWW类的使用不符合微软命名规范。

2>大量并发使用WWW类时会抛出Too Many Threads的异常.UnityHttpClient已经在内部对并发线程数量进行了控制。

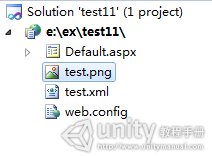
使用UnityHttpClient类,代码更加简洁。

下面通过代码展示如何使用UnityHttpClient类:

1.下载Warensoft Unity3D通信库

该类库是Codeplex上的开源项目,地址是:http://wucl.codeplex.com,从上面下载类库的最新版本,下载后得到两个DLL文件,其中Warensoft.Unity.Communication.dll就是我们在本Demo中用的DLL文件,另外一个Warensoft.DataService.dll是数据服务库,在后面的章节中会讲解如何使用.

2.为了客户端代码能成功演示,下面在VisualStudio中建立一个网站,当做Http服务器,为了演示UnityHttpClient对文件的请求,需要在网站中添加一个名为Test.xml的XML文件,以及一个test.png的PNG图,如下图所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/1.png)

Unity3D教程：利用UnityHttpClient类实现Http通信

Test.xml的内容如下所示:

<?xml version="1.0" encoding="utf-8" ?>

<Root>

<Customer>ALFKI</Customer>

</Root>

Test.png的图片如下所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/2.png)

Unity3D教程：利用UnityHttpClient类实现Http通信

3.为了演示通过Get方式请求数据,需要在网站中添加一个GetTest.aspx页面,其对应的ASPX.CS中的代码如下所示:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

public partial class GetTest : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//该代码功能是通过查询字符串CustomerID的值去查询该客户对应的公司名,

//并将查找到的公司返回给客户端

string customerid = this.Request.QueryString["CustomerID"];

if (customerid!=null )

{

string companyName = "";

using (SqlConnection con=new SqlConnection ("server=.;database=northwind;uid=sa;pwd=sa"))

{

var cmd = con.CreateCommand();

cmd.CommandText = "select companyname from customers where customerid=@customerid";

cmd.Parameters.AddWithValue("@customerid",customerid);

con.Open();

var result = cmd.ExecuteScalar();

if (result !=null )

{

companyName = result.ToString();

}

}

this.Response.Write(companyName);

this.Response.End();

}

}

}

4.为了演示通过POST方式发送数据,添加一个PostTest.aspx,其对应的CS文件代码如下所示:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

public partial class PostTest : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//该代码功能是通过获取客户端以Post方式发送的CustomerID的值去查询该客户//对应的公司名,并将查找到的公司返回给客户端

if (this.Request .ContentLength!=0)

{

//获取POST的数据流

var strm = this.Request.InputStream;

//建立缓冲区

byte[]buffer=new byte[this.Request .ContentLength];

//将POST过来的数据读取出来,并且存储在buffer中

strm.Read(buffer,0,buffer .Length );

//将二进制数据转化为字符串

string customerid = System.Text.Encoding.UTF8.GetString(buffer);

string companyName = "";

using (SqlConnection con = new SqlConnection("server=.;database=northwind;uid=sa;pwd=sa"))

{

var cmd = con.CreateCommand();

cmd.CommandText = "select companyname from customers where customerid=@customerid";

cmd.Parameters.AddWithValue("@customerid", customerid);

con.Open();

var result = cmd.ExecuteScalar();

if (result != null)

{

companyName = result.ToString();

}

}

this.Response.Write(companyName);

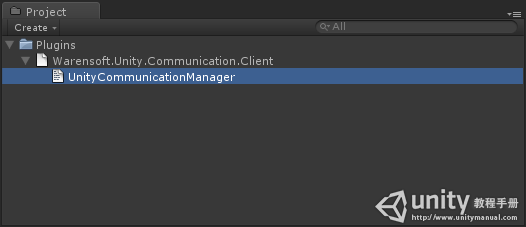
this.Response.End();

}

}

}

5.在Unity3D中建立一个新的项目,建立一个文件夹改名为Plugins,并将Warensoft.Unity.Communication.dll拷贝到该文件夹下面,如下图所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/3.png)

Unity3D教程：利用UnityHttpClient类实现Http通信

6.在Project中添加一个CS脚本,并将其命名为HttpTest.cs,其代码如下所示:

using UnityEngine;

using System.Collections;

using Warensoft.Unity.Communication.Client;

using System;

using System.Xml;

public class HttpTest : MonoBehaviour {

UnityHttpClient httpClient = null;

private Texture2D image;

void Start () {

//启动时将httpClient初始化

this.httpClient = UnityCommunicationManager.CreateInstance().GetHttpClient();

this.httpClient.Error += new EventHandler<HttpRequestErrorEventArgs>(httpClient\_Error);

}

void httpClient\_Error(object sender, HttpRequestErrorEventArgs e)

{

print(e.ResponseText );

}

int initStep = 0;

void Update () {

if (this.initStep ==0&&this.httpClient!=null )

{

//获取XML文件

this.httpClient.BeginGetHttpContent("http://localhost:17737/test11/test.xml",

//后面的方法是本次异步Http请求成功并响应后的回调方法

new Action<XmlDocument>((doc) =>

{

//打印XML文件

print("response xml content:");

print(doc.OuterXml);

}));

//获取图片

this.httpClient.BeginGetHttpContent("http://localhost:17737/test11/test.png",

//后面的方法是本次异步Http请求成功并响应后的回调方法

new Action<Texture2D>((img) =>

{

this.image = img;

}));

//获取纯文本

//通过客户ID查询公司名

//GET方式

this.httpClient.BeginGetHttpContent("http://localhost:17737/test11/GetTest.aspx?CustomerID=ALFKI",

//后面的方法是本次异步Http请求成功并响应后的回调方法

new Action<string>((stringResult) =>

{

//打印公司名

print("Get the company name of alfki:" + stringResult);

}));

//获取纯文本

//通过客户ID查询公司名

//POST方式

byte[] contentBuffer = System.Text.Encoding.UTF8.GetBytes("ALFKI");

this.httpClient.BeginPost("http://localhost:17737/test11/PostTest.aspx", contentBuffer,

(response) =>

{

//打印公司名

print("Post the company name of alfki:" + response.StringContent);

});

this.initStep = 1;

}

}

void OnGUI()

{

if (this.image !=null )

{

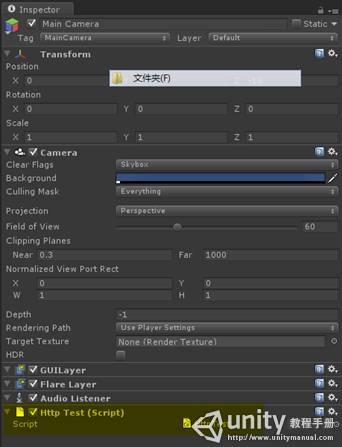
GUI.DrawTexture(new Rect (0,0,this.image .width,this.image.height),this.image);

}

}

}

7.将该cs文件拖放到该场景的主摄像机上,如下图所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/4.png)

Unity3D教程：利用UnityHttpClient类实现Http通信

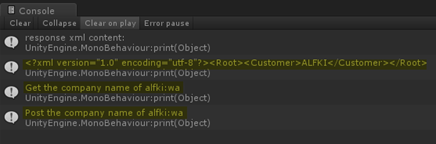
8.保存项目并运行

游戏视图输出如下图所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/5.png)

Unity3D教程：利用UnityHttpClient类实现Http通信

控制台输出内容如下图所示:

[](http://www.unitymanual.com/wp-content/uploads/2013/02/6.png)

Unity3D教程：利用UnityHttpClient类实现Http通信