**Unity3D教程：引用第三方DLL解析JSON**

Posted on 2013年07月16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 102 次

在日常的网络应用与游戏开发中，我们经常需要在服务端和客户端之间做数据交互。比如以前很火的社交游戏《开心农场》，当你进入好友家查看对方的菜有没有成熟的时候，客户端就开始向服务端请求该好友家的信息。服务端收到请求后就会将该玩家的农场资料发回给客户端，并由客户端解析后显示在游戏界面上。而中间的“语言”我们称之为数据交互格式。

目前页游中比较流行的有XML/JSON/AMF等等，当然也可以自己实现一套序列化/反序列化的解决方案。但使用这些久经考验并且现成的格式，可以省下很多重复造轮子的精力和时间。JSON.Net是我用得比较顺手的JSON序列化/反序列库，它是开源的。点击这里可以进入它的官方页面并下载。在写这篇笔记的时候最新版本为4.5R10，支持 .Net2.0/3.5/4.0，还有Silverlight/WP/Win8。

下载成功后可以得到一个zip压缩包，里边包含了源代码(Source目录)和编译好的DLL(Bin目录)，运行环境都已经按目录名分好了。这里有一点要注意，虽然Unity中使用.Net Framework 3.5作为工程的类库输出，但是不能使用Net35下面的DLL，而应该使用Net20下面的DLL。

所有由Unity引用的外部DLL，都必须放置在工程Assets/Plugins文件夹中（在Unity的工程面板中为Plugins目录），支持存放到子级文件夹。如左图所示，我在Plugins下面新建了一个Newtonsoft.Json的目录，并把DLL文件都扔进去。这样可以使得工程面板的目录结构更清晰一点，但并不是强制性的。

然后新建一个Test.cs脚本做测试，看看引用是否成功。别忘了脚本必须附加到场景上的某个物体上才会被调用。

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 | **using** Newtonsoft.Json; |
| 03 |  |
| 04 |  |
| 05 |  |
| 06 | **public** **class** TestData |
| 07 | { |
| 08 | **public** **string** PlayerName { **get**; **set**; } |
| 09 | **public** **int** Level { **get**; **set**; } |
| 10 | **public** **int** Exp { **get**; **set**; } |
| 11 | **public** **override** **string** ToString() |
| 12 | { *//Unity3D教程手册：www.unitymanual.com* |
| 13 | **return** **string**.Format("PlayerName={0},Level={1},Exp={2}", PlayerName, Level, Exp); |
| 14 | } |
| 15 | } |
| 16 |  |
| 17 | **public** **class** Test : MonoBehaviour |
| 18 | { |
| 19 | **void** Start() |
| 20 | { |
| 21 | **var** data = new TestData |
| 22 | { |
| 23 | PlayerName = "黄旭杨", |
| 24 | Level = 10, |
| 25 | Exp = 123, |
| 26 | }; |
| 27 | **var** serializedData = JsonConvert.SerializeObject(data); |
| 28 | Debug.Log("Serialized:" + serializedData); |
| 29 | **var** deserializedData = JsonConvert.DeserializeObject<TestData>(serializedData); |
| 30 | Debug.Log("Deserialized:" + deserializedData); |
| 31 | } |
| 32 | } |