**C# web api返回类型设置为json的两种方法**

作者：佚名 来源：我爱学习网 更新时间：2014-2-13

分享到

[1](http://www.poluoluo.com/jzxy/201402/262277.html)

web api写api接口时默认返回的是把你的对象序列化后以XML形式返回，那么怎样才能让其返回为json呢，下面就介绍两种方法：   
方法一：(改配置法)   
  
找到Global.asax文件，在Application\_Start()方法中添加一句：

。代码如下:

GlobalConfiguration.Configuration.Formatters.XmlFormatter.SupportedMediaTypes.Clear();

修改后：

。代码如下:

protected void Application\_Start()   
{   
AreaRegistration.RegisterAllAreas();   
WebApiConfig.Register(GlobalConfiguration.Configuration);   
FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);   
RouteConfig.RegisterRoutes(RouteTable.Routes);   
// 使api返回为json   
GlobalConfiguration.Configuration.Formatters.XmlFormatter.SupportedMediaTypes.Clear();   
}

这样返回的结果就都是json类型了，但有个不好的地方，如果返回的结果是String类型，如123，返回的json就会变成"123";   
  
解决的方法是自定义返回类型(返回类型为HttpResponseMessage)

。代码如下:

public HttpResponseMessage PostUserName(User user)   
{   
String userName = user.userName;   
HttpResponseMessage result = new HttpResponseMessage { Content = new StringContent(userName,Encoding.GetEncoding("UTF-8"), "application/json") };   
return result;   
}

方法二：(万金油法)   
该方法已验证  
方法一中又要改配置，又要处理返回值为String类型的json，甚是麻烦，不如就不用web api中的的自动序列化对象，自己序列化后再返回

。代码如下:

public HttpResponseMessage PostUser(User user)   
{   
JavaScriptSerializer serializer = new JavaScriptSerializer();   
string str = serializer.Serialize(user);   
HttpResponseMessage result = new HttpResponseMessage { Content = new StringContent(str, Encoding.GetEncoding("UTF-8"), "application/json") };   
return result;   
}

方法二是我比较推荐的方法，为了不在每个接口中都反复写那几句代码，所以就封装为一个方法这样使用就方便多了。

。代码如下:

public static HttpResponseMessage toJson(Object obj)   
{   
String str;   
if (obj is String ||obj is Char)   
{   
str = obj.ToString();   
}   
else   
{   
JavaScriptSerializer serializer = new JavaScriptSerializer();   
str = serializer.Serialize(obj);   
}   
HttpResponseMessage result = new HttpResponseMessage { Content = new StringContent(str, Encoding.GetEncoding("UTF-8"), "application/json") };   
return result;   
}

方法三：(最麻烦的方法)   
  
方法一最简单，但杀伤力太大，所有的返回的xml格式都会被毙掉，那么方法三就可以只让api接口中毙掉xml，返回json   
  
先写一个处理返回的类：

。代码如下:

public class JsonContentNegotiator : IContentNegotiator   
{   
private readonly JsonMediaTypeFormatter \_jsonFormatter;   
  
public JsonContentNegotiator(JsonMediaTypeFormatter formatter)   
{   
\_jsonFormatter = formatter;   
}   
  
public ContentNegotiationResult Negotiate(Type type, HttpRequestMessage request, IEnumerable<MediaTypeFormatter> formatters)   
{   
var result = new ContentNegotiationResult(\_jsonFormatter, new MediaTypeHeaderValue("application/json"));   
return result;   
}   
}

找到App\_Start中的WebApiConfig.cs文件，打开找到Register(HttpConfiguration config)方法   
  
添加以下代码：

。代码如下:

var jsonFormatter = new JsonMediaTypeFormatter();   
config.Services.Replace(typeof(IContentNegotiator), new JsonContentNegotiator(jsonFormatter));

添加后代码如下：

。代码如下:

public static void Register(HttpConfiguration config)   
{   
config.Routes.MapHttpRoute(   
name: "DefaultApi",   
routeTemplate: "api/{controller}/{action}/{id}",   
defaults: new { id = RouteParameter.Optional }   
);   
var jsonFormatter = new JsonMediaTypeFormatter();   
config.Services.Replace(typeof(IContentNegotiator), new JsonContentNegotiator(jsonFormatter));   
}

方法三如果返回的结果是String类型，如123，返回的json就会变成"123"，解决方法同方法一。   
  
其实web api会自动把返回的对象转为xml和json两种格式并存的形式，方法一与方法三是毙掉了xml的返回，而方法二是自定义返回。