**.net通过url访问服务器获取服务器返回数据**

原创 2016年08月15日 11:26:44

* 标签：
* [http协议](http://so.csdn.net/so/search/s.do?q=http%E5%8D%8F%E8%AE%AE&t=blog) /
* [https协议](http://so.csdn.net/so/search/s.do?q=https%E5%8D%8F%E8%AE%AE&t=blog) /
* [服务器返回数据](http://so.csdn.net/so/search/s.do?q=%E6%9C%8D%E5%8A%A1%E5%99%A8%E8%BF%94%E5%9B%9E%E6%95%B0%E6%8D%AE&t=blog) /
* [调用url](http://so.csdn.net/so/search/s.do?q=%E8%B0%83%E7%94%A8url&t=blog) /
* [获取服务器数据](http://so.csdn.net/so/search/s.do?q=%E8%8E%B7%E5%8F%96%E6%9C%8D%E5%8A%A1%E5%99%A8%E6%95%B0%E6%8D%AE&t=blog)
* 1383

一、url为http协议

1、普通调用：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. **public** **string** GetInfo(**string** url)
2. {
3. //访问http方法
4. **string** strBuff = "";
5. Uri httpURL = **new** Uri(url);
6. ///HttpWebRequest类继承于WebRequest，并没有自己的构造函数，需通过WebRequest的Creat方法建立，并进行强制的类型转换
7. HttpWebRequest httpReq = (HttpWebRequest)WebRequest.Create(httpURL);
8. ///通过HttpWebRequest的GetResponse()方法建立HttpWebResponse,强制类型转换
9. HttpWebResponse httpResp = (HttpWebResponse)httpReq.GetResponse();
10. ///GetResponseStream()方法获取HTTP响应的数据流,并尝试取得URL中所指定的网页内容
11. ///若成功取得网页的内容，则以System.IO.Stream形式返回，若失败则产生ProtoclViolationException错误。在此正确的做法应将以下的代码放到一个try块中处理。这里简单处理
12. Stream respStream = httpResp.GetResponseStream();
13. ///返回的内容是Stream形式的，所以可以利用StreamReader类获取GetResponseStream的内容，并以
14. //StreamReader类的Read方法依次读取网页源程序代码每一行的内容，直至行尾（读取的编码格式：UTF8）
15. StreamReader respStreamReader = **new** StreamReader(respStream, Encoding.UTF8);
16. strBuff = respStreamReader.ReadToEnd();
17. **return** strBuff;
18. }

方法调用举例：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. **string** url = "http://192.168.1.38:8088/api/user/get\_info";    //http协议url
2. url +="?token=" + token + "&openid=" + openid ;
3. **string** userInfo = GetInfo(url);

2、json调用（使用Json方法访问url需要在.net3.5以上版本）：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. #region Json访问url方式(使用该方法需要在.net3.5以上)
2. //获得用户信息
3. **protected** OAuthUser Get\_UserInfo(**string** token , **string** openid)
4. {
5. //http协议url
6. **string** Str = GetJson("http://192.168.1.38:8088/api/user/get\_info?token=" + token + "&openid=" + openid);
7. OAuthUser OAuthUser\_Model = JsonHelper.ParseFromJson(Str);
8. **return** OAuthUser\_Model;
9. }
11. **protected** **string** GetJson(**string** url)
12. {
13. //访问http
14. WebClient wc = **new** WebClient();
15. wc.Credentials = CredentialCache.DefaultCredentials;
16. wc.Encoding = Encoding.UTF8;
17. **string** returnText = wc.DownloadString(url);
18. **if** (returnText.Contains("errcode"))
19. {
20. //可能发生错误
21. }
22. //Response.Write(returnText);
23. **return** returnText;
24. }
25. ///
26. /// 用户信息类
27. ///
28. **public** **class** OAuthUser
29. {
30. **public** OAuthUser()
31. { }
32. #region 数据库字段
33. **private** **string** \_dept;
34. **private** **string** \_name;
35. **private** **string** \_userid;
36. #endregion
38. #region 字段属性
39. ///
40. /// 部门
41. ///
42. **public** **string** dept
43. {
44. **set** { \_dept = value; }
45. **get** { **return** \_dept; }
46. }
47. ///
48. /// 用户名称
49. ///
50. **public** **string** name
51. {
52. **set** { \_name = value; }
53. **get** { **return** \_name; }
54. }
55. ///
56. /// 用户的唯一标识
57. ///
58. **public** **string** userid
59. {
60. **set** { \_userid = value; }
61. **get** { **return** \_userid; }
62. }
63. #endregion
64. }
66. **public** **class** JsonHelper
67. {
68. ///
69. /// 生成Json格式
70. ///
71. ///
72. ///
73. ///
74. **public** **static** **string** GetJson(T obj)
75. {
76. DataContractJsonSerializer json = **new** DataContractJsonSerializer(obj.GetType());
77. **using** (MemoryStream stream = **new** MemoryStream())
78. {
79. json.WriteObject(stream, obj);
80. **string** szJson = Encoding.UTF8.GetString(stream.ToArray()); **return** szJson;
81. }
82. }
83. ///
84. /// 获取Json的Model
85. ///
86. ///
87. ///
88. ///
89. **public** **static** T ParseFromJson(**string** szJson)
90. {
91. T obj = Activator.CreateInstance();
92. **using** (MemoryStream ms = **new** MemoryStream(Encoding.UTF8.GetBytes(szJson)))
93. {
94. DataContractJsonSerializer serializer = **new** DataContractJsonSerializer(obj.GetType());
95. **return** (T)serializer.ReadObject(ms);
96. }
97. }
98. }
99. #endregion

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. json调用方式举例：
2. OAuthUser OAuthUser\_Model = Get\_UserInfo(token, openid);
3. //联调测试
4. Response.Write("accesstoken:" + token + "
5. userid:" + OAuthUser\_Model.userid + "
6. 用户名称:" + OAuthUser\_Model.name + "
7. 部门:" + OAuthUser\_Model.dept);

二、url为https协议  
只需加入一句话即可：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. //访问https方法（先调用以下方法）
2. **using** System.Security.Cryptography.X509Certificates;

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. **public** **bool** CheckValidationResult(**object** sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors errors)
2. {
3. //直接确认，否则打不开
4. **return** **true**;
5. }

1、普通调用：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. **public** **string** GetInfo(**string** url)
2. {
3. //访问https方法需加上这句话（CheckValidationResult就是以上方法）
4. ServicePointManager.ServerCertificateValidationCallback =
5. **new** System.Net.Security.RemoteCertificateValidationCallback(CheckValidationResult);
6. //访问http方法（不需要加上面那句话）
7. **string** strBuff = "";
8. Uri httpURL = **new** Uri(url);
9. ///HttpWebRequest类继承于WebRequest，并没有自己的构造函数，需通过WebRequest的Creat方法建立，并进行强制的类型转换
10. HttpWebRequest httpReq = (HttpWebRequest)WebRequest.Create(httpURL);
11. ///通过HttpWebRequest的GetResponse()方法建立HttpWebResponse,强制类型转换
12. HttpWebResponse httpResp = (HttpWebResponse)httpReq.GetResponse();
13. ///GetResponseStream()方法获取HTTP响应的数据流,并尝试取得URL中所指定的网页内容
14. ///若成功取得网页的内容，则以System.IO.Stream形式返回，若失败则产生ProtoclViolationException错误。在此正确的做法应将以下的代码放到一个try块中处理。这里简单处理
15. Stream respStream = httpResp.GetResponseStream();
16. ///返回的内容是Stream形式的，所以可以利用StreamReader类获取GetResponseStream的内容，并以
17. //StreamReader类的Read方法依次读取网页源程序代码每一行的内容，直至行尾（读取的编码格式：UTF8）
18. StreamReader respStreamReader = **new** StreamReader(respStream, Encoding.UTF8);
19. strBuff = respStreamReader.ReadToEnd();
20. **return** strBuff;
21. }

方法调用举例：

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. **string** url = "https://192.168.1.38:8088/api/user/get\_info";    //htts协议url
2. url +="?token=" + token + "&openid=" + openid ;
3. **string** userInfo = GetInfo(url);

2、Json调用

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. //获得用户信息
2. **protected** OAuthUser Get\_UserInfo(**string** token , **string** openid)
3. {
4. //htts协议url
5. **string** Str = GetJson("https://192.168.1.38:8088/api/user/get\_info?token=" + token + "&openid=" + openid);
6. OAuthUser OAuthUser\_Model = JsonHelper.ParseFromJson(Str);
7. **return** OAuthUser\_Model;
8. }
10. **protected** **string** GetJson(**string** url)
11. {
12. //访问https需加上这句话
13. ServicePointManager.ServerCertificateValidationCallback = **new** System.Net.Security.RemoteCertificateValidationCallback(CheckValidationResult);
14. //访问http（不需要加上面那句话）
15. WebClient wc = **new** WebClient();
16. wc.Credentials = CredentialCache.DefaultCredentials;
17. wc.Encoding = Encoding.UTF8;
18. **string** returnText = wc.DownloadString(url);
20. **if** (returnText.Contains("errcode"))
21. {
22. //可能发生错误
23. }
24. //Response.Write(returnText);
25. **return** returnText;
26. }

**[csharp]** [view plain](http://blog.csdn.net/angle_greensky110/article/details/52209497) [copy](http://blog.csdn.net/angle_greensky110/article/details/52209497)

1. json调用方式举例：
2. OAuthUser OAuthUser\_Model = Get\_UserInfo(token, openid);
3. //联调测试
4. Response.Write("accesstoken:" + token + "
5. userid:" + OAuthUser\_Model.userid + "
6. 用户名称:" + OAuthUser\_Model.name + "
7. 部门:" + OAuthUser\_Model.dept);