

推酷

- [文章](#)
- [站点](#)
- [主题](#)
- [活动](#)
- [公开课](#)
- [APP](#) [荐](#)
- [周刊](#)
 - [编程狂人](#)
 - [设计匠艺](#)
 - [创业周刊](#)
 - [科技周刊](#)
 - [Guru Weekly](#)
 - [一周拾遗](#)

不用找了, 比较全的 signalR例子已经为你准备好了.

[• 登录](#)

时间 2015-03-30 22:46:00 [博客园-原创精华区](#)

原文 <http://www.cnblogs.com/codefish/p/4371780.html>

主题 [SignalR 集线器 HTML](#)

这几天想着将一个winform的工具上线到web上, 因为对时时性的要求比较高, 找朋友咨询了一下推荐了SignalR 框架, 比较强大. 昨天才看到, 今天研究了一下将里面的例子都拿出来共享.

最全的参考:<http://www.asp.net/signalr/overview/getting-started>

关于如果安装SignalR: NuGet命令:

```
PM> Install-Package Microsoft.AspNet.SignalR
```

<-----1:与他人聊天:<-----

后台代码示例:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Microsoft.AspNet.SignalR;
using Microsoft.AspNet.SignalR.Hubs;
namespace SignalRChat.Hubs.Data
{
    [HubName("ViewDataHub")]
}
```

```

public class ViewDataHub : Hub
{
    //this is just called by client and return the value for it .
    public string Hello()
    {
        return "hello";
    }
    //this fuction will be called by client and the inside function
    //Clients.Others.talk(message);
    //will be called by client javascript function .
    public void SendMessag(string message)
    {
        Clients.Others.talk(message);
    }
}

```

小提示:注意其它的红色字体部分

前台代码示例:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <!--Script references. -->
    <!--Reference the jQuery library. -->
    <script src="Scripts/jquery-1.10.2.min.js"></script>
    <!--Reference the SignalR library. -->
    <script src="Scripts/jquery.signalR-2.0.2.js"></script>
    <!--Reference the autogenerated SignalR hub script. -->
    <script src='signalr/hubs'></script>
    <!--Add script to update the page and send messages.-->
    <script type='text/javascript'>
        $(function () {
            // Declare a proxy to reference the hub.
            var chat = $.connection.ViewDataHub;
            //init the client function
            init(chat);
            $("#btnclick").click(function () {
                //Response the information
                $.connection.hub.start().done(function () {
                    chat.server.hello().done(function (res) {
                        alert(res);
                    })//end of done function
                })//the end of the $.connection
            })//end of click function
            $("#btntalk").click(function () {
                $.connection.hub.start().done(function () {
                    chat.server.sendMessag($("#txttalk").val());
                    $("#txttalk").val("");
                })//the end of the $.connection
            })//btntalk end
        })
        //init the client method
        function init(chat) {
            chat.client.talk = function (message) {
                var talk = "<h1>" + message + "</h1>";
                $("#dvtalk").append(talk);
            } //end regist the client function
        } //end of the initfunction
    </script>
</head>
<body>
    <div>
        <table id="tbtoday"></table>
        <input type="text" id="txttalk" width="150"/>
        <input type="button" id="btnclick" value="clickme" />
        <input type="button" id="btntalk" value="talkwithme" />
        <div id="dvtalk">

```

```

    </div>
  </div>
</body>
</html>

```

出现的效果:



两个窗口之间的聊天

我知道你心中肯定有疑问,我也是这样,当我刚接触的时候完全搞不懂这是为什么会这样,我们来回顾一次正常的聊天过程:

客户端A发送信息->服务端接收->服务端Post消息给客户端B

那我们重新拆分以上的方法来证明我们的猜想是否正确

```

$("#btntalk").click(function () {
    $.connection.hub.start().done(function () {
        chat.server.sendMessage($("#txttalk").val());
        $("#txttalk").val("");
    })//the end of the $.connection

});//btntalk end

```

chat.server.sendMessage(message) 从客户端调用了服务器的方法(服务器扮演的是中转站的角色).

此时的message 从客户端A发送给了服务端

那服务器就应该有这样的一个方法与之相对应

后台代码:

```

1  //this fucntion will be called by client and the inside function
2  //Clients.Others.talk(message);
3  //will be called by client javascript function .
4  public void SendMessage(string message)
5  {
6      Clients.Others.talk(message);
7  }

```

服务端接收到A发送来的message.

这个时候服务端将消息推送给客户端B

Clients.Others.talk(message);

这个时候客户端B应该有一个talk的方法来将消息显示出来

```

1  //init the client method
2  function init(chat) {

```

```

3
4     chat.client.talk = function (message) {
5         var talk = "<h1>" + message + "</h1>";
6
7         $("#dvtalk").append(talk);
8
9     } //end regist the client function
10
11 } //end of the initfunction

```

这个时候客户端B接收到消息, 用Js的方法显示出来消息. 一次通话就完成了.

<-----二, 客户端传递参数给服务端并从服务端得到返回值:<-----

前端代码:



```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4     <title></title>
5     <!--Script references. -->
6     <!--Reference the jQuery library. -->
7     <script src="Scripts/jquery-1.10.2.min.js"></script>
8     <!--Reference the SignalR library. -->
9     <script src="Scripts/jquery.signalR-2.0.2.js"></script>
10    <!--Reference the autogenerated SignalR hub script. -->
11    <script src='signalr/hubs'></script>
12    <!--Add script to update the page and send messages.-->
13    <script type='text/javascript'>
14
15
16        $(function () {
17            // Declare a proxy to reference the hub.
18            var chat = $.connection.ViewDataHub;
19
20
21            //init the client function
22            init(chat);
23
24
25            $("#btnclick").click(function () {
26                //Response the information
27                $.connection.hub.start().done(function () {
28                    chat.server.hello($("#txttalk").val()).done(function (res) {
29                        var talk = "<h1>" + res + "</h1>";
30
31                        $("#dvtalk").append(talk);
32                    })//end of done function
33                })//the end of the $.connection
34            })//end of click function
35
36
37
38            $("#btntalk").click(function () {
39                $.connection.hub.start().done(function () {
40                    chat.server.sendMessage($("#txttalk").val());
41                    $("#txttalk").val("");
42                })//the end of the $.connection
43
44            });//btntalk end
45
46        })
47
48
49        //init the client method

```

```

50     function init(chat) {
51
52         chat.client.talk = function (message) {
53             var talk = "<h1>" + message + "</h1>";
54
55             $("#dvtalk").append(talk);
56
57         } //end register the client function
58
59     } //end of the initfunction
60
61     </script>
62 </head>
63 <body>
64     <div>
65         <table id="tbtoday"></table>
66         <input type="text" id="txttalk" width="150"/>
67         <input type="button" id="btnclick" value="clickme" />
68         <input type="button" id="btntalk" value="talkwithme" />
69         <div id="dvtalk">
70
71         </div>
72     </div>
73 </body>
74 </html>

```

View Code

后端代码:



```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using Microsoft.AspNet.SignalR;
6 using Microsoft.AspNet.SignalR.Hubs;
7
8 namespace SignalRChat.Hubs.Data
9 {
10     [HubName("ViewDataHub")]
11     public class ViewDataHub : Hub
12     {
13         //this is just called by client and return the value for it .
14         public string Hello(string msg)
15         {
16
17             string res = "sorry , i don't know";
18             if (msg.Contains("hello"))
19             {
20                 res = "hello , guy";
21             }
22             if (msg.Contains("how"))
23             {
24                 res = "how are you ~";
25             }
26             if (msg.Contains("love"))
27             {
28                 res = "don't fall in love with me ~";
29             }
30             return res;
31         }
32
33
34
35         //this fuction will be called by client and the inside function
36         //Clients.Others.talk(message);
37         //will be called by clienet javascript function .
38         public void SendMessag(string message)

```

```

39      {
40          Clients.Others.talk(message);
41      }
42  }
43  }
44  }

```

View Code

效果图:



<-----三，客户端刷时时的刷新数据<-----

前端:



```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5     <title></title>
6     <script src="Scripts/jquery-1.6.4.min.js"></script>
7     <script src="Scripts/jquery.signalR-2.2.0.min.js"></script>
8     <script src='signalr/hubs'></script>
9     <script>
10         var chat;
11         var arr = [];
12         var loopdo;
13         //
14         $(function () {
15             // Created proxy
16             chat = $.connection.viewDataHub;
17             Init(chat);
18             //on the start done
19             $.connection.hub.start().done(function () {
20
21                 $("#btnloop").click(function () {
22                     chat.server.refresh().done(function (data) {
23                         AppendTable(data);
24                     })//done end
25                 }) //btnloop end
26
27                 $("#btnclient").click(function () {
28                     chat.server.RefreshClients();
29                 });
30
31                 $("#btnclick").click(function () {
32                     Start();
33                 });
34
35                 $("#btnstop").click(function () {
36                     Stop();
37                 });
38             })
39         }) //function
40
41         function Start() {
42             if (loopdo == null) {

```

```

48         loopdo = setInterval('$("#btnloop").click()', 1000);
49     }
50 }
51 }
52
53 function Stop() {
54     if (loopdo != null) {
55         clearInterval(loopdo);
56     }
57 }
58
59 function AppendTable(data) {
60     arr.length = 0;
61     arr.push("<tr><th>Opendoor</th><th>Price</th><th>Opentiem</th></tr>");
62     $.each(data, function (i) {
63         arr.push("<tr>");
64         arr.push("<td>" + data[i].Opendoor + "</td>");
65         arr.push("<td>" + data[i].Price + "</td>");
66         arr.push("<td>" + data[i].Opentiem + "</td>");
67         arr.push("</tr>");
68     });
69     $("#tblist").html(arr.join(""));
70 }
71
72 function Init(chat) {
73     chat.client.myrefresh = function (data) {
74         AppendTable(data);
75     }
76 }
77 }
78
79 </script>
80 </head>
81 <body>
82     <input type="button" id="btnclick" value="Start" />
83
84     <input type="button" id="btnloop" value="View" style="display: none" />
85
86     <input type="button" id="btnstop" value="Stop" />
87
88     <input type="button" id="btnclient" value="ClientAll" />
89     <table id="tblist">
90     </table>
91 </body>
92
93
94
95 </html>

```

View Code

后端:



```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using Microsoft.AspNet.SignalR;
6 using Microsoft.AspNet.SignalR.Hubs;
7 namespace MyResh.ViewData
8 {
9     [HubName("viewDataHub")]
10    public class ViewDataHub : Hub
11    {
12        [HubMethodName("refresh")]
13        public List<Stock> Refresh()
14        {
15            return Stock.GetAll();

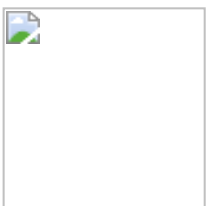
```

```
16     }
17
18     [HubMethodName("RefreshClients")]
19     public void RefreshClients()
20     {
21         Clients.All.myrefresh(Stock.GetAll());
22     }
23
24
25 }
26
27 public class Stock
28 {
29     private string opendoor;
30
31     public string Opendoor
32     {
33         get { return opendoor; }
34         set { opendoor = value; }
35     }
36
37     private double price;
38
39     public double Price
40     {
41         get { return price; }
42         set { price = value; }
43     }
44
45     private DateTime opentiem = System.DateTime.Now;
46
47     public DateTime Opentiem
48     {
49         get { return opentiem; }
50         set { opentiem = value; }
51     }
52
53
54     public static List<Stock> GetAll()
55     {
56         Random rand = new Random();
57         List<Stock> list = new List<Stock>()
58         {
59             new Stock{Opendoor="Door1",Price=rand.NextDouble()*100},
60             new Stock{Opendoor="Door2",Price=rand.NextDouble()*100},
61             new Stock{Opendoor="Door3",Price=rand.NextDouble()*100}
62         };
63         return list;
64     }
65
66 }
67 }
```

View Code

注：本次实现的所谓的时时刷新数据和官方提供的Demo有很大的差异，并不是后台代码的角度来刷新，而是从前端的角度来实现的。

[====>点我下载DEMO<====](#)



分享

收藏

纠错

handone




管理工具 一个就够

项目协作 客户管理 知识共享 流程审批...

推荐文章

- 1. [ASP.NET MVC——CodeFirst开发模式](#)
- 2. [\[爬虫学习笔记\]Url过滤模块UrlFilter](#)
- 3. [静态构造函数](#)
- 4. [C#异步编程](#)
- 5. [C# 泛型接口中的协变与抗变（逆变）](#)
- 6. [EF不能很好的支持DDD？估计是我们搞错了！](#)

相关推刊

- by [蓝色玫瑰](#)  《默认推刊》 1
- by [visionwang](#)  《默认推刊》 2278
- by [晏言_iz4](#)  《默认推刊》 13

我来评几句

请输入评论内容...

登录后评论

已发表评论数(0)

相关站点

[博客园-原创精华区](#)[+ 订阅](#)

热门文章

- 1. [ASP.NET MVC——CodeFirst开发模式](#)
- 2. [\[爬虫学习笔记\]Url过滤模块UrlFilter](#)
- 3. [静态构造函数](#)
- 4. [EF不能很好的支持DDD？估计是我们搞错了！](#)
- 5. [httpHandlers与Http处理程序](#)

handone

管理工具 一个就够

项目协作 客户管理 知识共享 流程审批...

接入 **DaoVoice**
千元大礼等你拿

活动

DaoCloud

90%的产品经理都在用

极光推送 **不只是稳定**

JIGUANG | 极光 JPUSH | 极光推送

简道云

零门槛搭建数据应用

7 阿里云 周年庆

9.20-9.30 **感恩特惠**

云服务器85折再送6个月

阿里云 230万用户信赖

广告

让性能不再阻挡 游戏的激情

立即性能优化

透视宝

teambition

国内领先的团队协作工具

访问页面被拦截

因为您没有权限访问该页面，
如有疑问请联系您的网络管理

收藏到推刊

[创建推刊](#)

推刊名(必填)

推刊描述

请填写推刊名

描述不能大于100个字符!

权限设置：☒ 公开 ☐ 仅自己可见



文章纠错

邮箱地址

错误类型

正文不准确

补充信息

网站相关

- [关于我们](#)
- [移动应用](#)
- [建议反馈](#)

关注我们



友情链接

- [人人都是产品经理](#)
- [PM256](#)
- [移动信息化](#)
- [行晓网](#)
- [智城外包网](#)
- [虎嗅](#)
- [IT耳朵](#)
- [创媒工场](#)
- [经理人分享](#)
- [市场部网](#)
- [砍柴网](#)
- [CocoaChina](#)
- [北风网](#)
- [云智慧](#)
- [我赢职场](#)
- [大数据时代](#)
- [奇笛网](#)
- [咕噜网](#)
- [红联linux](#)
- [Win10之家](#)
- [鸟哥笔记](#)
- [爱游戏](#)

[投资潮](#) [31会议网](#) [极光推送](#) [Teambition](#) [硅谷网](#) [leangoo](#) [ZEALER中国](#)
[OpenSNS](#) [小牛学堂](#) [handone](#) [Scrum中文网](#) [比戈大牛](#) [又拍云](#) [更多链接](#)
[>>](#)