vs2010 使用SignalR 提高B2C商城用户体验（一） - 鸟寅

时间 2014-04-25 18:16:00  [博客园-原创精华区](http://www.tuicool.com/sites/Fn2umm)

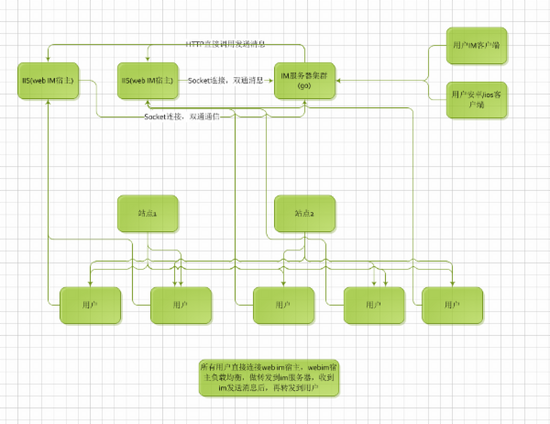
原文  [http://www.cnblogs.com/nywd/p/3689350.html](http://www.cnblogs.com/nywd/p/3689350.html?utm_source=tuicool&utm_medium=referral)

主题 [Visual Studio](http://www.tuicool.com/topics/11050018)[SignalR](http://www.tuicool.com/topics/11050022)

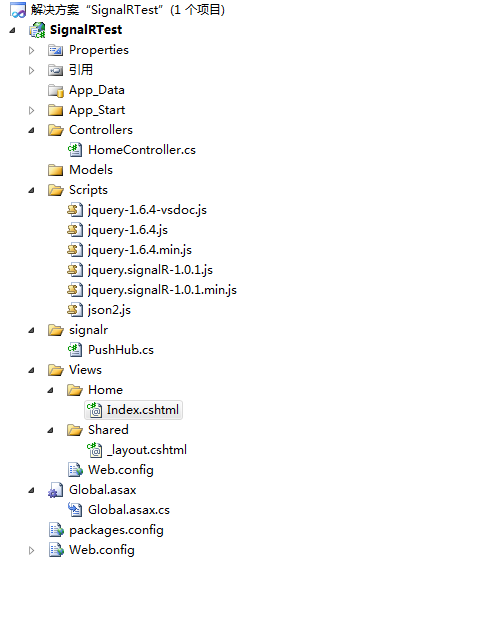
**vs2010 使用SignalR 提高B2C商城用户体验（一）**

1、需求简介，做为新时代的b2c商城，没有即时通讯，怎么提供用户粘稠度，怎么增加销量，用户购物的第一习惯就是咨询，即时通讯，应运而生。这里使用SignalR来实现即时通讯，再好不过了，不过项目依然简历在2010的基础上，所以不能使用新版本的SignalR了，这里介绍一下1.0.0版本的。

2、整个框架大概是这样搭建的，欢迎拍砖：



3、启动VS2010，新建一个项目，这里我建立的是asp.net mvc4



然后通过nuget引入SignalR所需要的组件

     Install-Package Microsoft.AspNet.SignalR -Version 1.0.1

关于nuget怎么使用就不过多介绍了，回车后，所需要的组件基本已经引入项目中了，下面进入开发。

（1）先介绍推送服务器开发：

在项目下，新加一个文件夹SignalR,

然后建立我们的集线器，

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** System.Web;

**using** Microsoft.AspNet.SignalR;

**using** Microsoft.AspNet.SignalR.Hubs;

**using** Owin;

**namespace** SignalRTest.SignalR

{

[HubName("pushHub")]

**public** **class** PushHub : Hub

{

**public** **void** Send(**string** message)

{

Clients.All.addMessage(message);

}

**public** **void** Bind(**string** userKey)

{

}

**public** **override** Task OnConnected()

{

**var** id = Context.ConnectionId;

**var** c = Clients.Caller;

**return** **base**.OnConnected();

}

}

}

Send方法，主要用来向客户端广播消息，Bind我本意是做登录校验，然后绑定客户端，OnConnected当客户端发起连接后会出发，里面的成员可以用一个字典进行管理。

然后进入Global.asax,配置一下路由，我这里的配置是适用于跨域的，后面会介绍到。

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Web;

**using** System.Web.Http;

**using** System.Web.Mvc;

**using** System.Web.Routing;

**using** Microsoft.AspNet.SignalR;

**using** Owin;

**namespace** SignalRTest

{

*// Note: For instructions on enabling IIS6 or IIS7 classic mode,*

*// visit http://go.microsoft.com/?LinkId=9394801*

**public** **class** MvcApplication : System.Web.HttpApplication

{

**protected** **void** Application\_Start()

{

**var** config = **new** HubConfiguration();

config.EnableCrossDomain = **true**;

RouteTable.Routes.MapHubs(config);

AreaRegistration.RegisterAllAreas();

WebApiConfig.Register(GlobalConfiguration.Configuration);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

}

}

}

（2）web页面开发：

先创建母版页，请注意js的顺序，json2.js主要是用于支持ie7及以下版本的。

1 @{

2 Layout = null;

3 }

4 **<!DOCTYPE html>**

5 <html>

6 <head>

7 <meta name="viewport" content="width=device-width" />

8 <title>singalr</title>

9 <script src="@Url.Content("~/Scripts/jquery-1.6.4.js")" type="text/javascript"></script>

10 <script src="@Url.Content("~/Scripts/json2.js")" type="text/javascript"></script>

11 <script src="@Url.Content("~/Scripts/jquery.signalR-1.0.1.js")" type="text/javascript"></script>

12 <script src="@Url.Content("~/signalr/hubs")" type="text/javascript"></script>

13 </head>

14 <body>

15 <div>

16 @RenderBody()

17 </div>

18 </body>

19 </html>

然后创建主页，我是在homecontroller里面创建的index，这里就不过多介绍了：

1 @{

2 Layout = "~/views/shared/\_layout.cshtml";

3 }

4 <script type="text/javascript">

5

6 $(**function** () {

7

8

9 *// Proxy created on the fly*

10 **var** chat = $.connection.pushHub;

11 *// Declare a function on the chat hub so the server can invoke it*

12 chat.client.addMessage = **function** ( message) {

13 writeEvent('<b>ny</b> 对大家说: ' + message, 'event-message');

14 };

15

16 $("#broadcast").click(**function** () {

17 *// Call the chat method on the server*

18 chat.server.send( $('#msg').val())

19 .done(**function** () {

20 console.log('Sent message success!');

21

22 })

23 .fail(**function** (e) {

24 console.warn(e);

25 });

26 });

27

28 *// Start the connection*

29 $.connection.hub.start().done(**function**() {

30 *//绑定*

31 chat.server.bind("wo");

32 });

33

34 *//A function to write events to the page*

35 **function** writeEvent(eventLog, logClass) {

36 **var** now = **new** Date();

37 **var** nowStr = now.getHours() + ':' + now.getMinutes() + ':' + now.getSeconds();

38 $('#messages').prepend('<li class="' + logClass + '"><b>' + nowStr + '</b> ' + eventLog + '.</li>');

39 }

40 });

41 </script>

42 <h2>

43 Hub Chat</h2>

44 <div>

45 <input type="text" id="Placeholder" value="ny" hidden="true" />

46 <input type="text" id="msg" />

47 <input type="button" id="broadcast" value="广播" />

48 <br />

49 <br />

50 <h3>

51 消息记录： (你是：<span id="MyClientName">ny</span>):

52 </h3>

53 <ul id="messages">

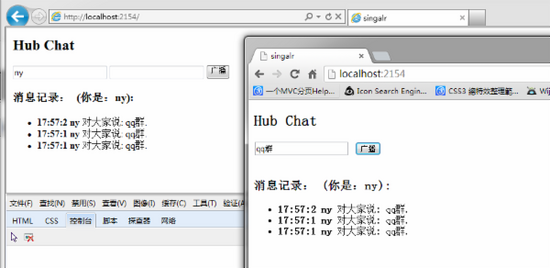
54 </ul>

55 </div>

此时，启动我们的项目



已经可以进行广播了，开启多个客户端看一下效果



是不是感觉很不错。此时，只是当前系统的，并不能做到与其他系统互通有无，那我们加个web方法，提供给其他服务器进行推送调用：

1 **using** System;

2 **using** System.Collections.Generic;

3 **using** System.Linq;

4 **using** System.Web;

5 **using** System.Web.Mvc;

6 **using** Microsoft.AspNet.SignalR;

7 **using** Microsoft.AspNet.SignalR.Hubs;

8 **using** SignalRTest.SignalR;

9

10 **namespace** SignalRTest.Controllers

11 {

12 **public** **class** HomeController : Controller

13 {

14 *//*

15 *// GET: /Home/*

16 */// <summary>*

17 */// 用户界面*

18 */// </summary>*

19 */// <returns></returns>*

20 **public** ActionResult Index()

21 {

22 **return** View();

23 }

24

25 */// <summary>*

26 */// web接口，推送消息*

27 */// </summary>*

28 */// <param name="msg"></param>*

29 */// <returns></returns>*

30 **public** **string** Send(**string** msg)

31 {

32 **var** context = GlobalHost.ConnectionManager.GetHubContext<PushHub>();

33 context.Clients.All.addMessage(msg);

34 **return** "ok";

35 }

36

37 }

38 }

这时我们来调用一下看看：



这样，我们就可以给客服人员做一个页面，让他们来给指定用户，或者全站用户广播消息了，是不是感觉很棒。

今天先到这里，要去面试个人，到这里，我们的推送服务器基本搭建完成，可以在这个基础上进行扩展，做出一个很棒的推送服务器，下一章我们会介绍跨域通信，与其他子站通讯，还有与客户端通信。