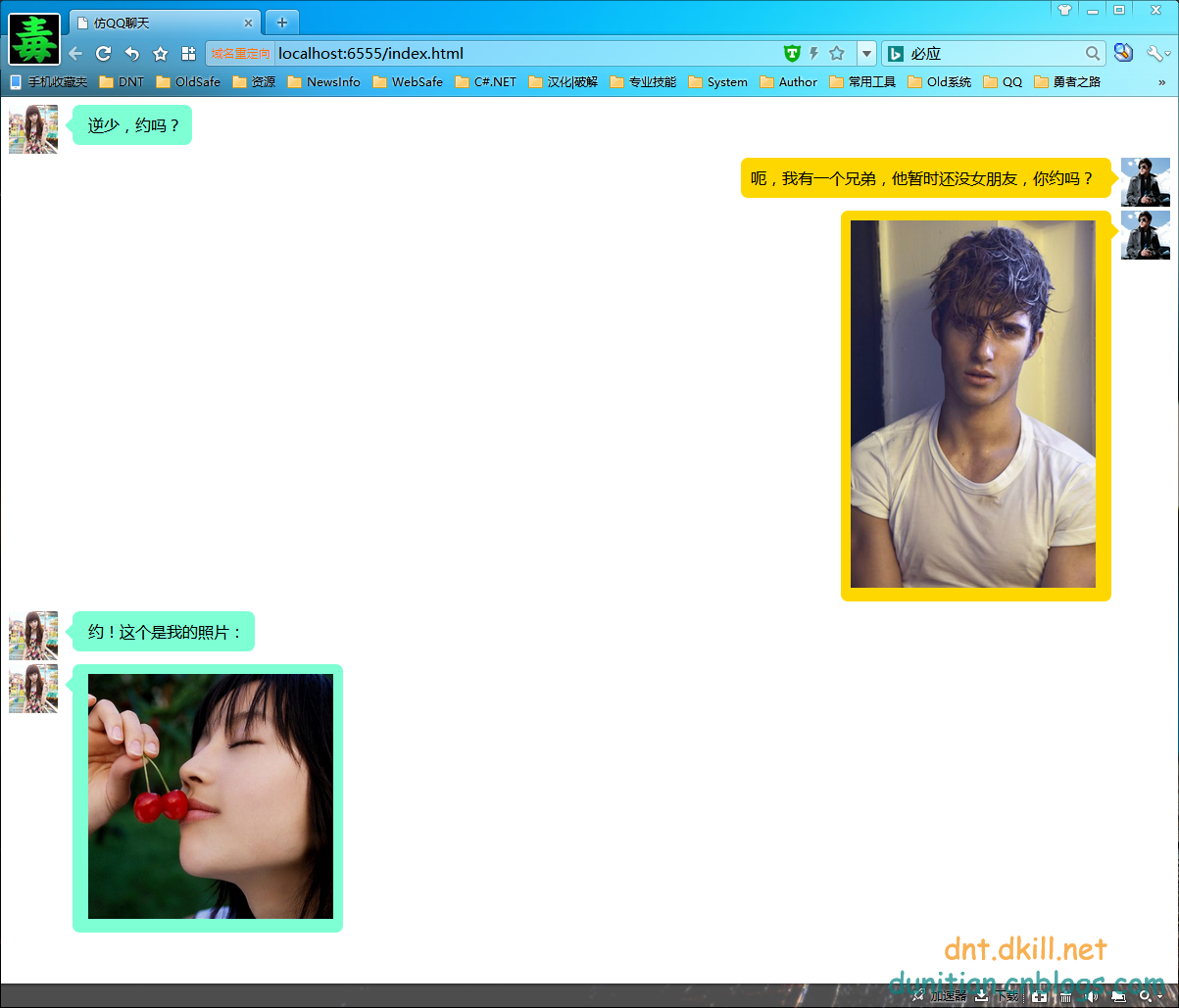
[SignalR快速入门 ~ 仿QQ即时聊天，消息推送，单聊，群聊，多群公聊（基础=》提升）](http://www.cnblogs.com/dunitian/p/5226515.html)

**SignalR快速入门 ~ 仿QQ即时聊天，消息推送，单聊，群聊，多群公聊（基础=》提升，5个Demo贯彻全篇，感兴趣的玩才是真的学）**

**官方demo**:<http://www.asp.net/signalr/overview/getting-started/tutorial-getting-started-with-signalr>

应用情景之一：



没太多连续的时间来研究SignalR，所以我把这篇文章分了三个阶段：

第一个阶段，简单使用，熟悉并认识SignalR

第二个阶段，实现上图的单聊效果

第三个阶段，实现类似QQ群发的功能

扩展阶段，如果有时间，逆天会再开一篇，封装一个LoTSignalR，看过逆天封装的人都知道，绝对简单又轻量级

比如[LoTQQ](http://www.cnblogs.com/dunitian/p/5055789.html)，现在已经很多人在用了，后期会添加新功能，敬请期待~~

++++++++++++++++++ **我是华丽的分割线**+++++++++++++++++++++

步入正轨：

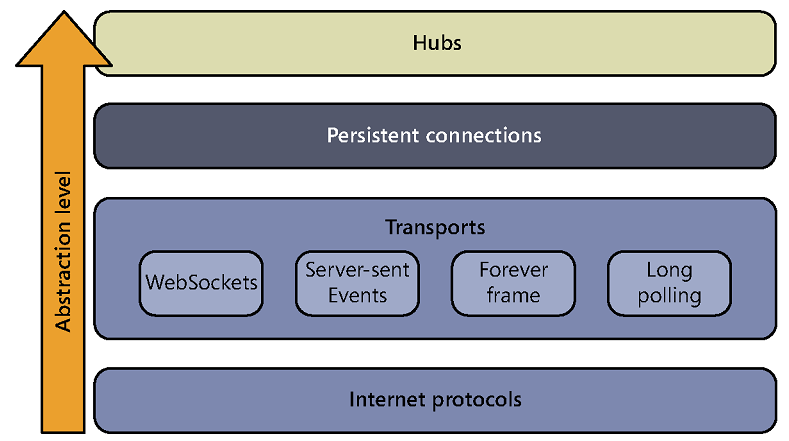
第一个阶段：

**1.什么是ASP.NET SignalR？**

**ASP .NET SignalR是一个 ASP .NET 下的类库，可以在ASP .NET 的Web项目中实现实时通信。什么是实时通信的Web呢？就是让客户端（Web页面）和服务器端可以互相通知消息及调用方法，当然这是实时操作的。**

**WebSockets是Html5提供的新的API，可以在Web网页与服务器端间建立Socket连接，当WebSockets可用时（即浏览器支持Html5）SignalR使用WebSockets，当不支持时SignalR将使用其它技术来保证达到相同效果。**

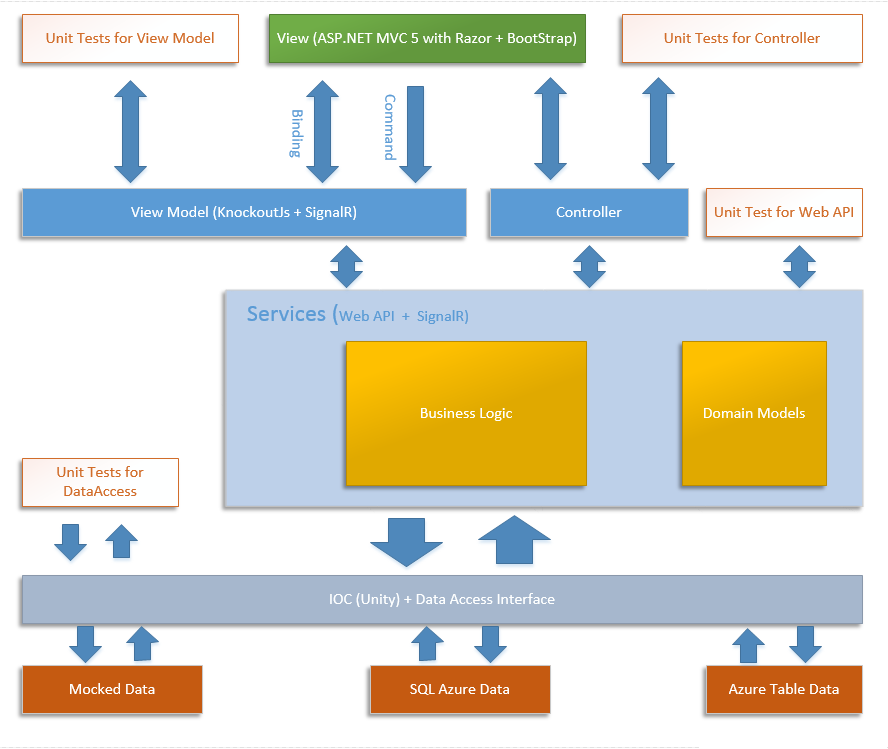
**SignalR当然也提供了非常简单易用的高阶API，使服务器端可以单个或批量调用客户端上的JavaScript函数，并且非常 方便地进行连接管理，例如客户端连接到服务器端，或断开连接，客户端分组，以及客户端授权，使用SignalR都非常 容易实现。**

****

**2.可以使用ASP.NET SingalR做什么？**  
**SignalR 将与客户端进行实时通信带给了ASP .NET 。当然这样既好用，而且也有足够的扩展性。以前用户需要刷新页面或使用Ajax轮询才能实现的实时显示数据，现在只要使用SignalR，就可以简单实现了。**

**最重要的是您无需重新建立项目，使用现有ASP .NET项目即可无缝使用SignalR。**

**网上某架构图：**

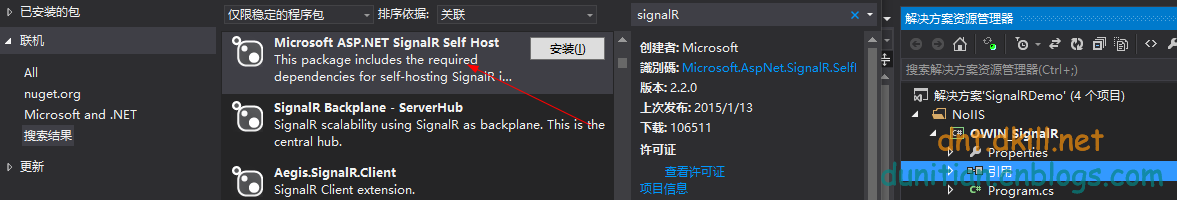


上面乱七八糟的估计很多人懒得看，好吧，你可以这样理解：

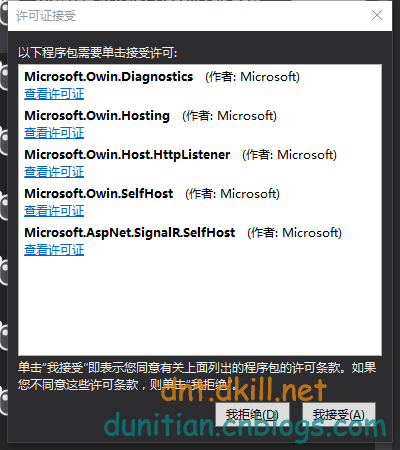
**使用了SignalR就可以让客户端通过SignalR代理直接调用服务端的方法，让服务端通过SignalR直接调用客户端的方法**

下面我们来实例演示一下，先演示一下不用IIS的情况：

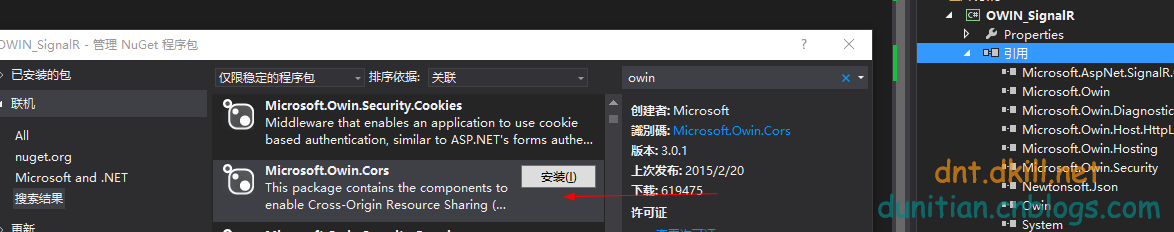
新建一个控制台项目，引入**signalR Self Host （**可以思考一下为什么会用NuGet包，他到底好在哪？O(∩\_∩)O~不清楚等项目演示完你应该就知道了**）**



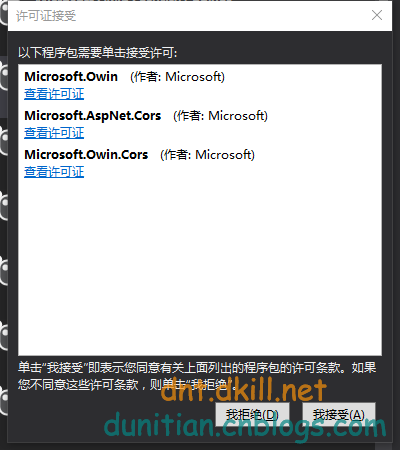
这是他的依赖项



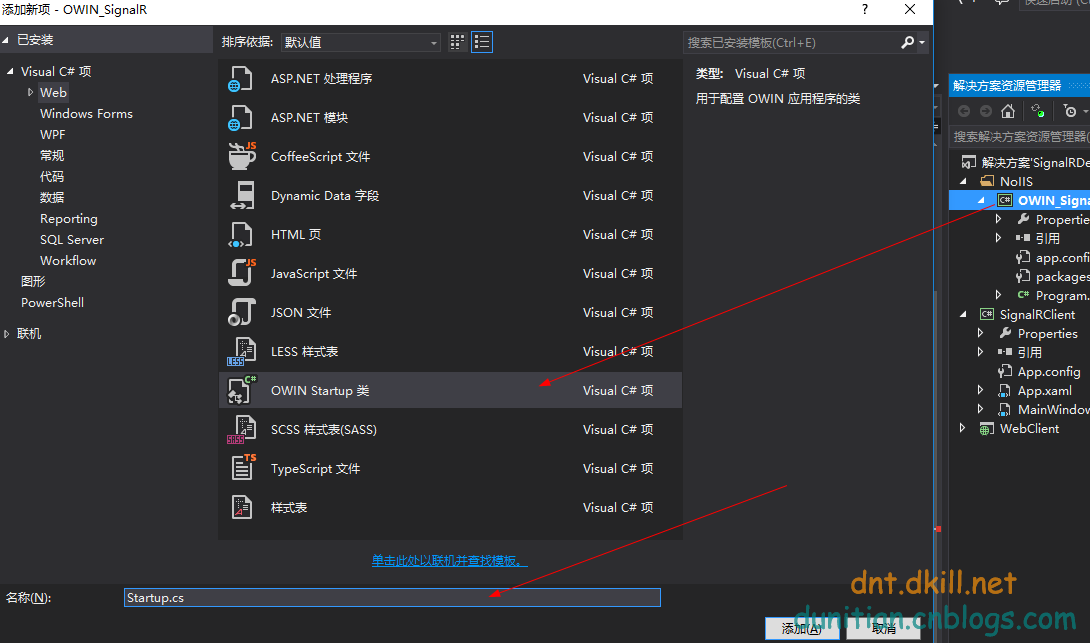
安装一下Owin.Cors



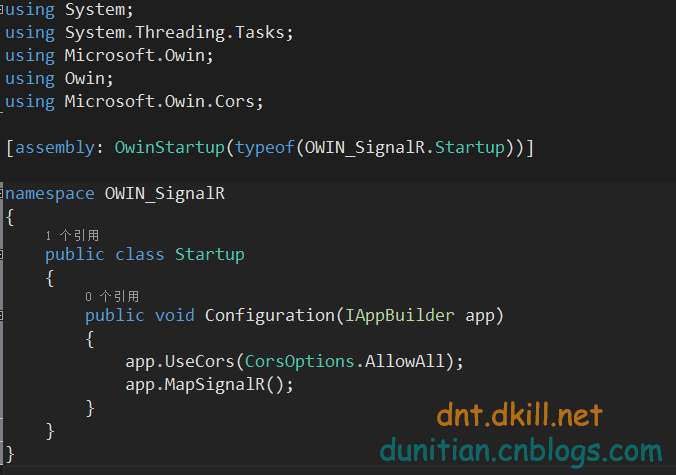
依赖项：



新增一个Owin的Startup类，类似于我们传统项目的Global文件



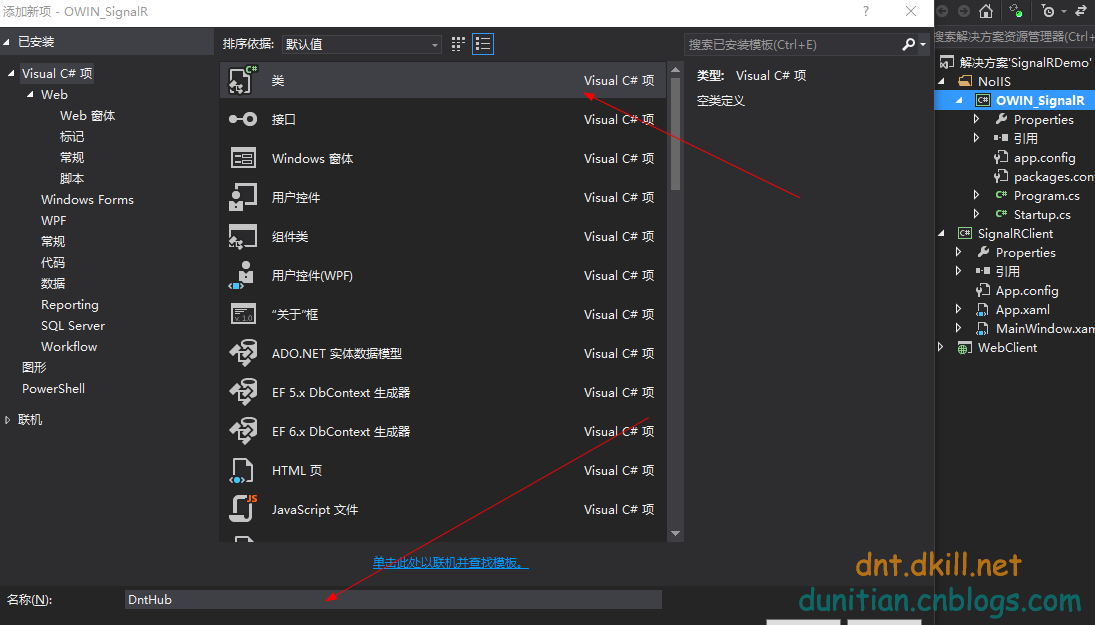
注册一下signalR中间组件（学过mvc的可以变相的理解为注册路由之类的）



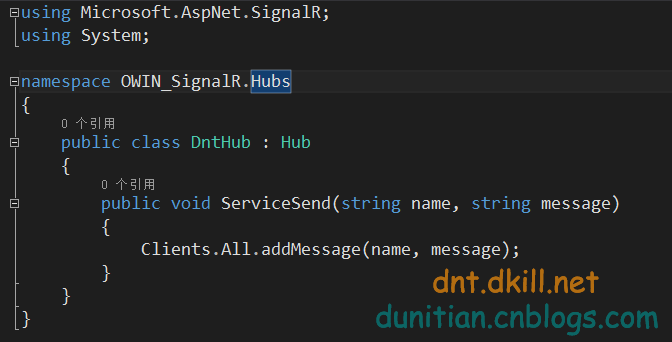
在main方法中绑定端口（不一定是8080，比如我demo中就用的其他端口）



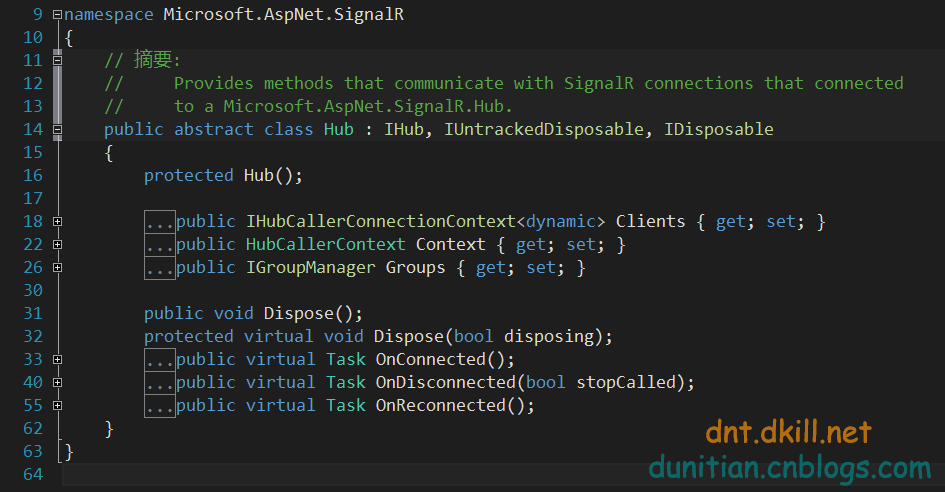
创建一个“SignalR集线器”（控制台这边因为没有集线器所以只能自己建类）



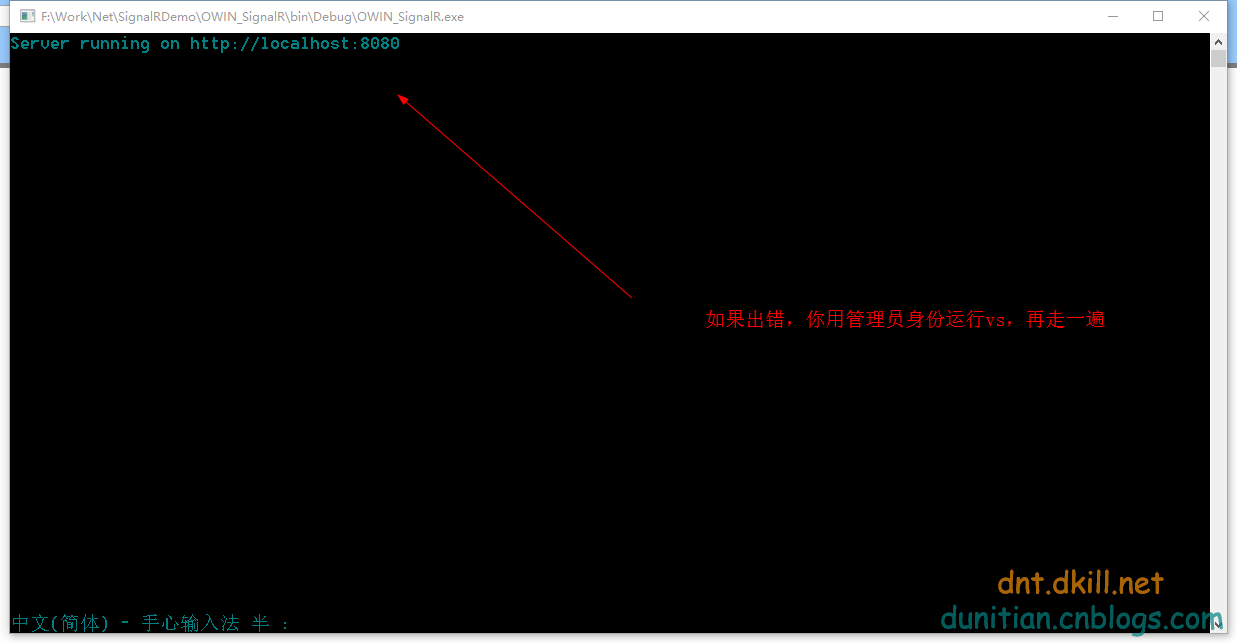
建了一个DntHub，定义了一个服务器端的方法，叫ServiceSend（一会会用到）



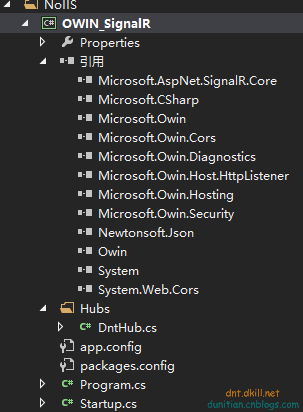
转到定义，看看Hub类（好东西啊，还有分组啥的，下面会讲）



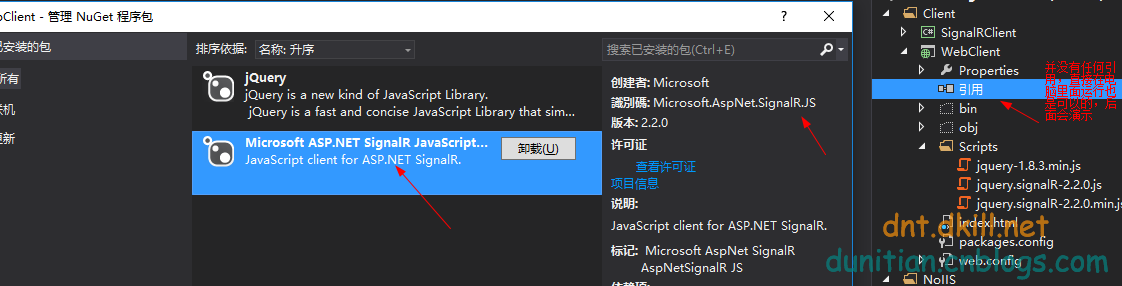
运行一下，如果出错请参考我的这篇文章：<http://www.cnblogs.com/dunitian/p/5232229.html>



先看看共引用多少dll（**和MVC之类的比起是不是简洁很多？一会演示好处在哪**）最小引用



创建一个web的Client，引用一下 SignalR的js包，其实你会发现就是多了几个js包并没有引用任何dll（必须的，不然岂不是太臃肿？）



前端的调用步骤：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <!--  总结一下：  1.先引入jq包，再引入signalR的js包，再引入signalR动态生成的hubs  2.设置signalR的hubs url地址：$.connection.hub.url =xxx  3.声明一个代理对象来引用集线器：var chat = $.connection.dntHub;  4.创建一个客户端方法：chat.client.xxxx=function(){}  5.启动并调用服务端方法：      $.connection.hub.start().done(function(){          chat.server.xxx()      });  --> |

代码贴起：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>简单聊天程序</title>

<style type="text/css">

.container {

background-color: #99CCFF;

border: thick solid #808080;

padding: 20px;

margin: 20px;

}

</style>

</head>

<body>

<div class="container">

<input type="text" id="message" />

<input type="button" id="sendmessage" value="Send" />

<input type="hidden" id="displayname" />

<ul id="discussion"></ul>

</div>

<script src="Scripts/jquery-1.8.3.min.js"></script>

<script src="Scripts/jquery.signalR-2.2.0.js"></script>

<!--动态生成的-->

<script src="http://localhost:5438/signalr/hubs"></script>

<script type="text/javascript">

$(function () {

//日记记录

$.connection.hub.logging = true;

//设置hubs的url

$.connection.hub.url = "http://localhost:5438/signalr";

// 声明一个代理来引用该集线器。

var chat = $.connection.dntHub;

// 创建一个方法供服务端调用

chat.client.addMessage = function (name, message) {

var encodedName = $('<div />').text(name).html();

var encodedMsg = $('<div />').text(message).html();

$('#discussion').append('<li><strong>' + encodedName + '</strong>:&nbsp;&nbsp;' + encodedMsg + '</li>');

};

$('#displayname').val('路人');

// 启动 connection

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

$('#sendmessage').click(function () {

//调用服务器端方法

chat.server.serviceSend($('#displayname').val(), $('#message').val());

});

});

});

</script>

<!--

总结一下：

1.先引入jq包，再引入signalR的js包，再引入signalR动态生成的hubs

2.设置signalR的hubs url地址：$.connection.hub.url =xxx

3.声明一个代理对象来引用集线器：var chat = $.connection.dntHub;

4.创建一个客户端方法：chat.client.xxxx=function(){}

5.启动并调用服务端方法：

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

chat.server.xxx()

});

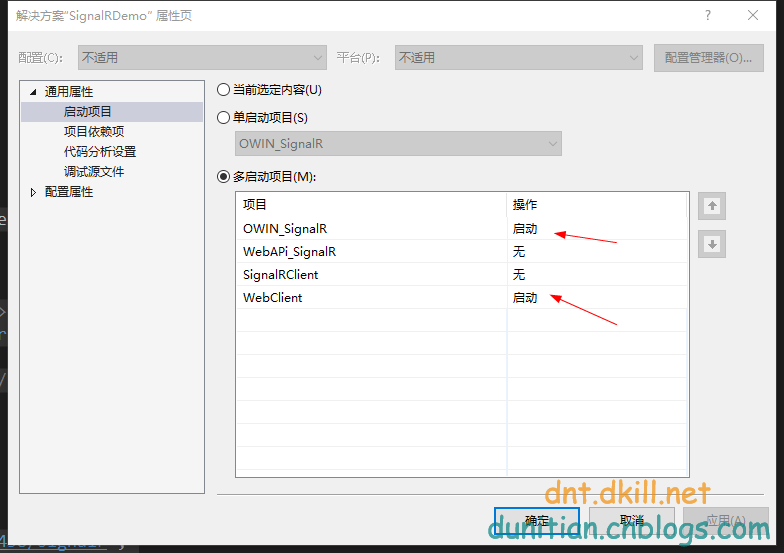
-->

</body>

</html>

[复制代码](javascript:void(0);)

如果要调试的话，保证服务端先运行，调试小技巧：



下面说一下上面的好处：

webclient我就单独拿出来了（一个js包，一个index.html），控制台的程序我也单独拿出来了，下面先运行一下服务端，再打开index.html



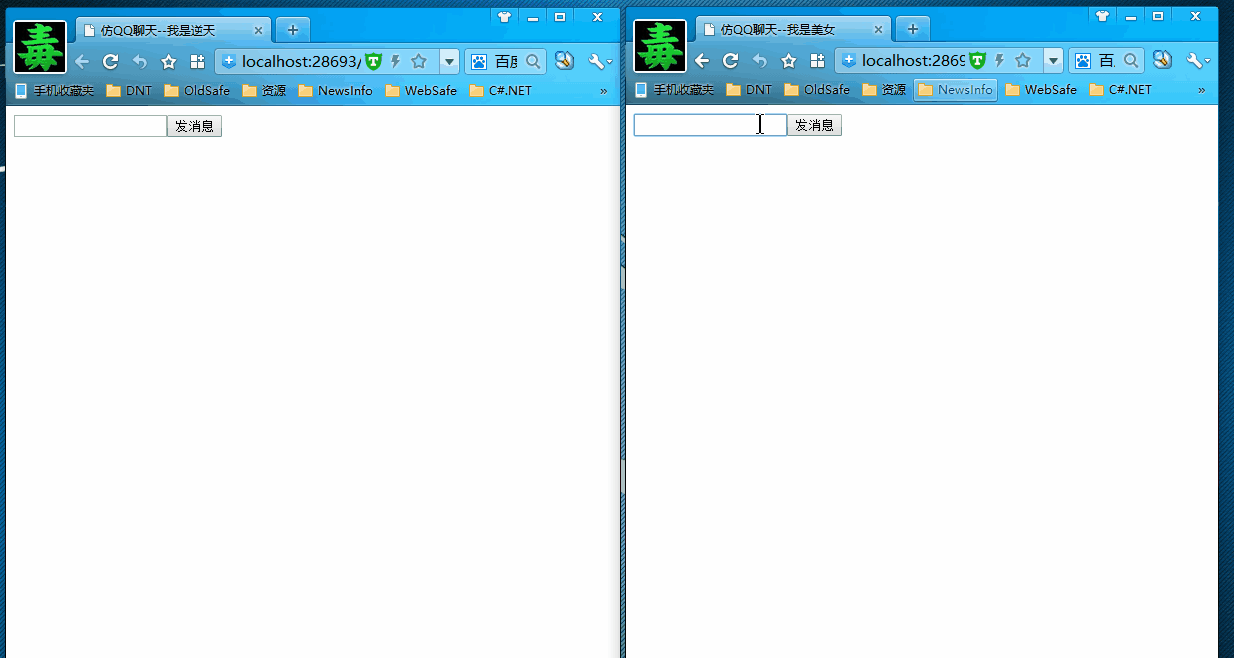
再次验证最上面说的，端口不固定



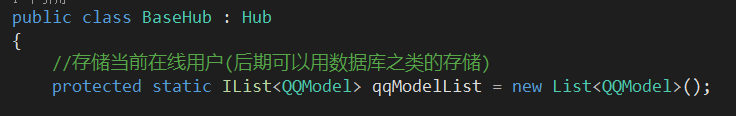
第二个阶段：

 QQ聊天案例，先讲一种常规的方法，下面会讲一种简单方法

先看看gif效果图把



定义一个BaseHub类，里面用 qqModeList来临时存放数据（用户数据）



QQModel，目前就用到两个属性，其他的可以自己扩展



定义了一个上线方法，一会每个客户端都会调用（本来是准备用OnConnected的，没办法他没参数。。。而且这个时候，qq昵称还没有产生，于是我取其次的方案）



定义一个发消息的方法



下面就是前端的东西了，注释很详细，不清楚可以直接留言，我没高兴深度封装，主要就是简单演示一下

代码贴上：

客户端-逆天

|  |
| --- |
| <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>仿QQ聊天--我是逆天</title>  <link href="Style/MyQQ.css" rel="stylesheet" />  </head>  <body>  <div><input id="inputMsg" /><input id="btn" type="button" value="发消息" /></div><br /><br />  <div id="main"></div>  <script src="Scripts/jquery-2.2.1.min.js"></script>  <script src="Scripts/jquery.signalR-2.2.0.min.js"></script>  <script src="http://localhost:5438/signalr/hubs"></script>  <script type="text/javascript">  $(function () {  var leftHtml = [  '<div class="sender">',  '<div><img src="http://img.jfdown.com/jfdown/201403/ygald3wajct.jpg"></div>',  '<div><div class="left\_triangle"></div>',  '<span>{msg}</span></div></div>'  ].join('');  var rightHtml = [  '<div class="receiver">',  '<div><img src="http://tb.himg.baidu.com/sys/portrait/item/306c9328?t=1397975854"></div>',  '<div><div class="right\_triangle"></div>',  '<span>{msg}</span></div></div>'  ].join('');  //设置hubs的url  $.connection.hub.url = 'http://localhost:5438/signalR';  // 声明一个代理  var qqProxy = $.connection.qQHub;  // 创建一个方法供服务端调用  qqProxy.client.sendMsg = function (msg) {  $('#main').append(leftHtml.replace('{msg}', msg));  }  // 启动 connection  $.connection.hub.start().done(function () {  qqProxy.server.online('逆天');//QQ昵称  $('#btn').click(function () {  //获取输入  var qqmsg = $('#inputMsg').val();  //给逆天发消息  qqProxy.server.serviceSend('妹子', qqmsg);  $('#main').append(rightHtml.replace('{msg}', qqmsg));  });  });  });  </script>  </body>  </html> |

客户端-美女

|  |
| --- |
| <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>仿QQ聊天--我是美女</title>  <link href="Style/MyQQ.css" rel="stylesheet" />  </head>  <body>  <div><input id="inputMsg" /><input id="btn" type="button" value="发消息" /></div><br /><br />  <div id="main"></div>  <script src="Scripts/jquery-2.2.1.min.js"></script>  <script src="Scripts/jquery.signalR-2.2.0.min.js"></script>  <script src="http://localhost:5438/signalr/hubs"></script>  <script type="text/javascript">  $(function () {  var rightHtml = [  '<div class="sender">',  '<div><img src="http://img.jfdown.com/jfdown/201403/ygald3wajct.jpg"></div>',  '<div><div class="left\_triangle"></div>',  '<span>{msg}</span></div></div>'  ].join('');  var leftHtml = [  '<div class="receiver">',  '<div><img src="http://tb.himg.baidu.com/sys/portrait/item/306c9328?t=1397975854"></div>',  '<div><div class="right\_triangle"></div>',  '<span>{msg}</span></div></div>'  ].join('');  //设置hubs的url  $.connection.hub.url = 'http://localhost:5438/signalR';  // 声明一个代理  var qqProxy = $.connection.qQHub;  // 创建一个方法供服务端调用  qqProxy.client.sendMsg = function (msg) {  $('#main').append(leftHtml.replace('{msg}', msg));  }  // 启动 connection  $.connection.hub.start().done(function () {  qqProxy.server.online('妹子');//QQ昵称  $('#btn').click(function () {  //获取输入  var qqmsg = $('#inputMsg').val();  //给逆天发消息  qqProxy.server.serviceSend('逆天', qqmsg);  $('#main').append(rightHtml.replace('{msg}', qqmsg));  });  });  });  </script>  </body>  </html> |

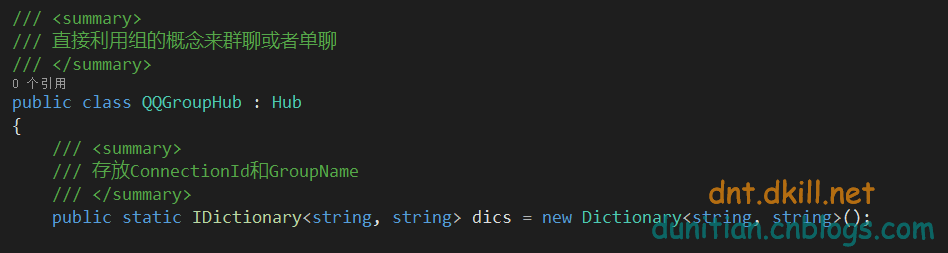
第三阶段

 群发

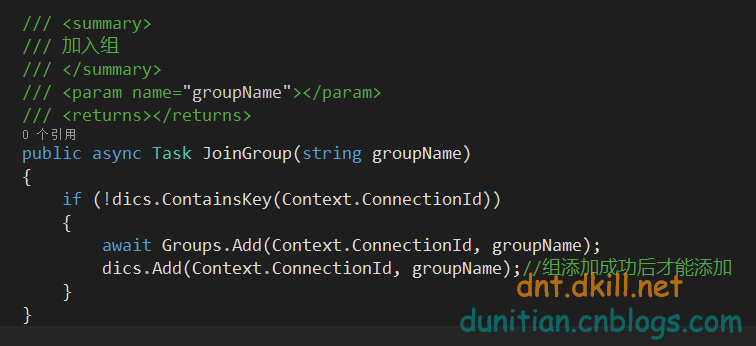
我们先接着昨天的QQ聊天来说

这次用一个简单的方法搞定

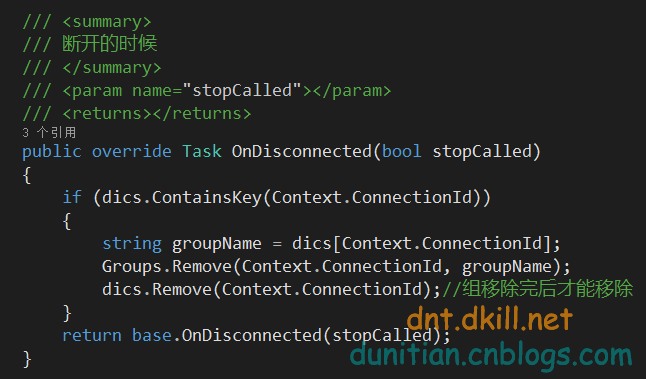
一个键值对集合存放ConnectionId和GroupName



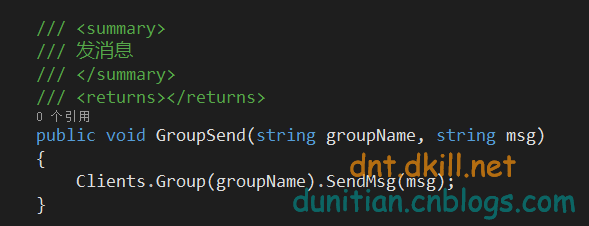
每个客户端连接的时候会加入一个组



 断开的时候退出组



 发消息



前端也进行了优化。ok，signalR第三个demo诞生，比昨天简单多了

