

黄明辉的博客

ASP.NET SignalR 系列（四）之指定对象推送

在上一章讲到了广播推送，即所有订阅的用户都能收到，这种适合于信息广播。

接下来介绍如何给指定的对象推送

在讲这个之前先说明一下连接创建的基础知识

- 1、每个页面与服务端创建连接并启动时，这时服务端会产生一个connectionId作为与这个客户端连接的唯一标识。
- 2、这个connectionId将作为服务端向指定客户端推送的依据
- 3、同一个页面刷新后ConnectionId一样不变
- 4、这个connectionId是服务端自动生成的一个随机数，无法变更。

一、只推送给自己

为了与之前的例子区分，我们分别在服务端DemoHub新建个发送给自己的方法




```
public class DemoHub : Hub
{
    /// <summary>
    /// 示例
    /// </summary>
    /// <param name="content">广播的内容</param>
    public void Hello(string content)
    {
        Clients.All.show(content); //调用了前端js上定义的hello方法
    }

    /// <summary>
    /// 推送给自己
    /// </summary>
    /// <param name="content"></param>
    public void CallSelf(string content)
    {
        Clients.Caller.show(content);
    }
}
```



这里使用了Caller，而不是All，表示只推送给呼叫方，即自己。

前端脚本逻辑不变，只是多了一个发送给自己的按钮的推送到服务端CallSelf事件而已。



```
//定义推送
$.connection.hub.start()
.done(function () {
    $("#btn_sendCall").click(function () {
        chat.server.callSelf($("#content").val()); //将客户端的content内容发送到服务端

        $("#content").val("");
    });
});
```

公告

昵称：黄明辉
园龄：6年1个月
粉丝：5
关注：1
[+加关注](#)

2020年6月						
日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[SignalR\(9\)](#)
[asp.net core\(6\)](#)
[Autofac\(4\)](#)
[MVC\(4\)](#)
[redis\(4\)](#)
[分布式锁\(3\)](#)
[前缀树\(3\)](#)
[SM\(2\)](#)
[国密\(2\)](#)
[ASP.NET\(2\)](#)
[更多](#)

随笔档案

[2020年4月\(1\)](#)
[2020年2月\(3\)](#)
[2019年12月\(3\)](#)
[2019年11月\(1\)](#)
[2019年8月\(1\)](#)
[2019年7月\(1\)](#)
[2019年6月\(7\)](#)
[2019年3月\(2\)](#)

2018年11月(3)
2018年9月(2)
2018年8月(9)

最新评论

- 1. Re:ASP.NET Core 在windows服务器安装dotnet-hosting-2.2.6-win 时出现应用程序池停止问题的解决方案
安装后还是不行啊

--夯
- 2. Re:ASP.NET 国密加密 SM2-SM4
@SM2把我折磨疯了 你的公钥必须得是SM2的，如果是RSA的，用不了 ...
--黄明辉
- 3. Re:ASP.NET 国密加密 SM2-SM4
@黄明辉 我有公钥，麻烦问一下应该如何用自己的公钥？ ...
--SM2把我折磨疯了
- 4. Re:ASP.NET 国密加密 SM2-SM4
@SM2把我折磨疯了 代码里面有生成公钥和私钥的方法，用那个方法生成密钥对...
--黄明辉
- 5. Re:ASP.NET 国密加密 SM2-SM4
@刘发源 不需要证书...
--黄明辉

阅读排行榜

- 1. ASP.NET 异步编程之Async await(1540)
- 2. 百度webuploader如何实现秒传与断点续传(1130)
- 3. ASP.NET SignalR 系列（六）之连接事件(924)
- 4. ASP.NET SignalR 系列（四）之指定对象推送(888)
- 5. ASP.NET SignalR 系列（八）之跨域推送(661)

评论排行榜

- 1. ASP.NET 国密加密 SM2-SM4(6)
- 2. ASP.NET MVC IOC 之 Autofac 系列开篇(1)
- 3. ASP.NET Core开发期间部署到IIS自定义主机域名并附加进程调试(1)
- 4. ASP.NET Core 在windows服务器安装dotnet-hosting-2.2.6-win 时出现应用程序池停止问题的解决方案(1)

推荐排行榜

- 1. ASP.NET SignalR 系列（四）之指定对象推送(2)
- 2. ASP.NET SignalR 系列（八）之跨域推送(1)
- 3. ASP.NET Core+MongoDB（一）(1)
- 4. ASP.NET SignalR 系列（三）之代码实现(1)

结果：



我是在窗口2这边发送的，第一条点击“发送”按钮，是广播，两个窗口都能收到；第二条点击“发送给自己”，只有窗口2收到。

二、其他推送方式

其他推送方式对于客户端（前端）来说不变，主要就是服务端推送对象改变而已

以下列出其他推送方式：

- 所有连接的客户端
`Clients.All.show(content);`
- 仅调用的客户端
`Clients.Caller.show(content);`
- 除调用客户端之外的所有客户端
`Clients.Others.show(content);`
- 特定的客户端标识的连接 id
`Clients.Client(Context.ConnectionId).show(content);`
- 所有连接的客户端除外指定客户端，由连接 ID 标识
`Clients.AllExcept(connectionId1, connectionId2).show(content);`
- 指定组中的所有连接的客户端
`Clients.Group(groupName).show(content);`

- 指定组中的所有连接的客户端除外指定客户端，由连接 ID 标识。
`Clients.Group(groupName, connectionId1, connectionId2).show(content);`
- 所有连接的客户端指定组中除调用客户端
`Clients.OthersInGroup(groupName).show(content);`
- 所有客户端和组列表中的连接 Id
`Clients.Clients(ConnectionIds).show(content);`
- 组的列表。
`Clients.Groups(GroupIds).show(content);`
- 按名称的用户
`Clients.Client(username).show(content);` //这里的username我还没理解是怎么创建的
- （在 SignalR 2.1 中引入）的用户名称的列表。
`Clients.Users(new string[] { "myUser", "myUser2" }).show(content)`

三、说说ConnectionId

前面已经说到，每个连接的页面都会产生不同的连接id，并且这个连接id是随机产生，不能自定义的，那我们如何获取呢

集线器中为我们提供了Context对象可以获取到这个连接id，先看下面简单的例子

```
/// <summary>
/// 返回每个连接的id
/// </summary>
public void ReturnConnectionId()
{
    Clients.Caller.show(Context.ConnectionId);
}
```

前端：

```
//定义推送,返回各个客户端连接
$.connection.hub.start()
    .done(function () {
        $("#btn_sendConnectionId").click(function () {
            chat.server.returnConnectionId(); //将客户端的content内容发送到服务端
        });
    });
```

结果：



从结果可以看出，分别点击返回各自的id，返回了不同的id。这个就验证了系统给每个调用端分配一个不相同的随机ConnectionId。

那我们如何利用这个ConnectionId来实现给某个客户端发送消息，比如一对一的聊天。这个就需要我们实现将人与ConnectionId信息建立对应关系。

如，我们假设张三和李四一对一聊天，那怎么实现，张三发送李四的消息只有李四收到呢，反之，亦然。即我们需要分别确定张三和李四的ConnectionId，并将它们的ConnectionId与张三、李四建立对应关系。

即建立以下的对应关系

张三 张三的ConnectionId

李四 李四的ConnectionId

张三和李四是已知的，那么就需要在张三和李四分别与集线器建立连接生成ConnectionId时，我们就要实现对应关系的建立。

这时候需要重载建立连接的方法：

服务端：



客户端：建立zhangsang.html和lisi.html 分别表示张三和李四的窗口

zhangsang.html

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
<link href="Content/bootstrap.min.css" rel="stylesheet" />
<script src="Scripts/jquery-1.10.2.min.js"></script>
<script src="Scripts/jquery.signalR-2.3.0.min.js"></script>
<script src="/signalr/hub/hubs"></script>
<meta charset="utf-8"/>
<style type="text/css">
    body {
        margin: 20px;
    }
    .input {
        padding-left: 5px;
    }
</style>
</head>
<body>
<div>
<h4>我是张三</h4>
<p>
    <input type="text" id="content" placeholder="" class="input"/> &nbsp;&nbsp;&nbsp;<input
type="button" value="发送给李四" class="btn btn-sm btn-info" id="btn_send"/>
</p>

<div>
<h4>接收到的信息: </h4>
<ul id="dataContainer">

</ul>
</div>
</div>

<script language="javascript">
    $(function() {
        var chat = $.connection.demoHub; //连接服务端集线器, demoHub为服务端集线器名称, js上首字母须
        改为小写 (系统默认)
        //定义客户端方法, 此客户端方法必须与服务端集线器中的方法名称、参数均一致。
        //实际上是服务端调用了前端的js方法 (订阅)

        $.connection.hub.qs = { 'userName': '张三' }
        chat.client.show=function(content) {
            var html = '<li>' + htmlEncode(content) + "</li>";
            $("#dataContainer").append(html);
        }

        //定义推送
        $.connection.hub.start()
            .done(function() {
                $("#btn_send").click(function() {
                    chat.server.callOne("李四", $("#content").val()); //将客户端的content内容发送
                    到服务端

                    $("#content").val("");
                });
            });
        //编码
        function htmlEncode(value) {
            var encodedValue = $('<div />').text(value).html();
            return encodedValue;
        }
    })
</script>
</body>
</html>

```


1、参数传递的方法

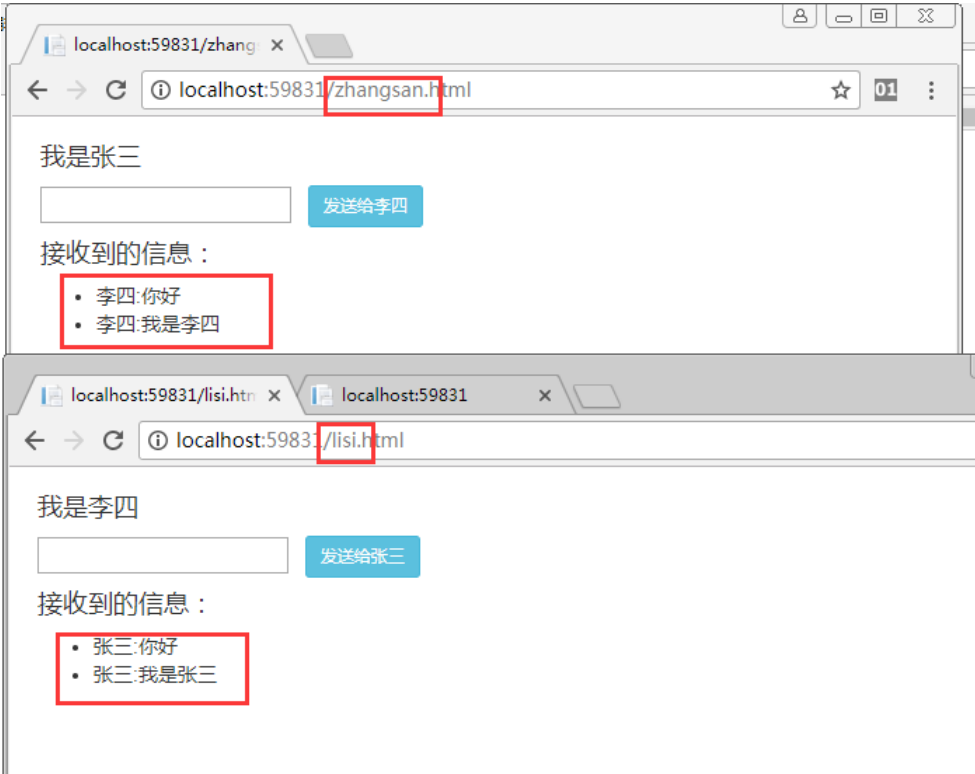
```
$.connection.hub.qs = { 'userName': '李四' }
```

2、后端获取参数的方法：

```
Context.QueryString["username"]
```

3、本例中利用Application对象来存储用户与ConnectionId的对应关系，当然，实际项目中也可以用其他方式存储。

示例结果：



本章结束！

标签: SignalR

好文要顶

关注我

收藏该文

黄明辉

关注 - 1

粉丝 - 5

+加关注

« 上一篇: [ASP.NET SignalR 系列（三）之代码实现](#)
» 下一篇: [ASP.NET SignalR 系列（五）之群组推送](#)

posted @ 2018-08-30 20:35 黄明辉 阅读(888) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】了解你才能更懂你，博客园首发问卷调查，助力社区新升级

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】Java实战200例（附源码）-助你提升实战能力

最新 IT 新闻:

- 戴尔发布全新商用系列电脑，让 AI 也来协助办公
 - 丁磊也直播卖货了，这届CEO都活成了“吉祥物”
 - 便利蜂宣布升级疫情防控 推出四大保障措施
 - 《率土之滨》大型抗议活动之下，网易游戏帝国的盛世危言
 - 有钱就放到余额宝里的人，这习惯恐怕要改一改了
- » 更多新闻...

历史上的今天:

- 2018-08-30 ASP.NET SignalR 系列（三）之代码实现
- 2018-08-30 ASP.NET SignalR 系列（二）之项目创建
- 2018-08-30 ASP.NET SignalR 系列（一）之SignalR介绍