

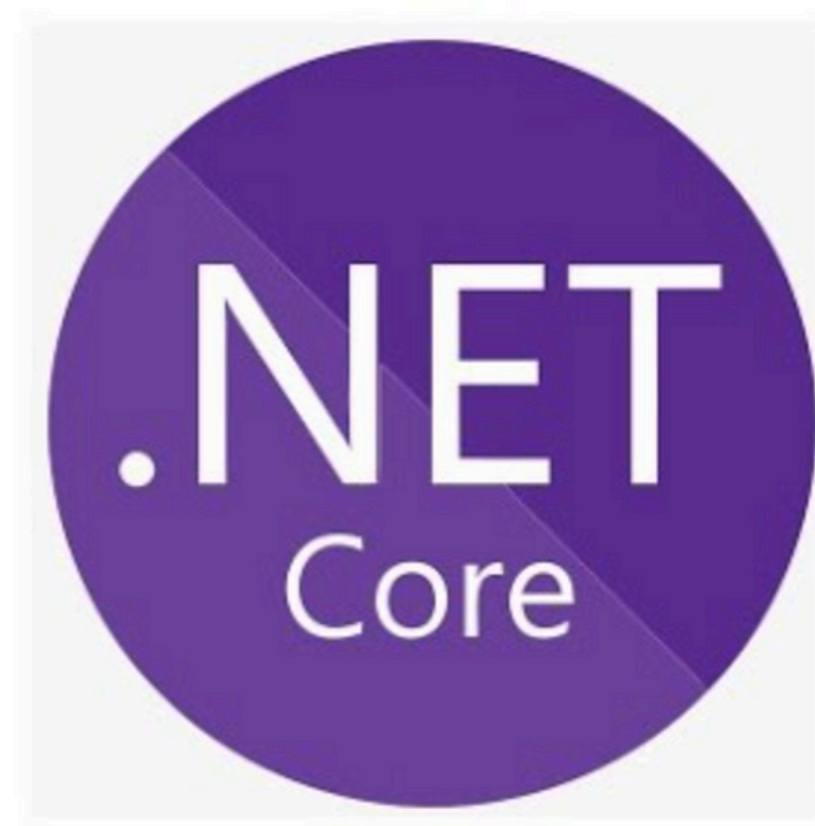
构建实时通信应用：使用.NET和SignalR轻松实现前沿技术！

作者：微信公众号：【架构师老卢】

11-28 14:18

154

SignalR



概述：学习.NET中使用SignalR实现实时通信功能。从安装库、创建Hub，到客户端基础功能，一步步构建实时聊天室。深入讲解分组功能，使您能够定向广播消息。简洁实用，助您轻松掌握实现创新Web应用的技能。

SignalR是一个强大的实时通信库，为.NET应用程序提供轻松的实时功能。它支持双向通信，让服务器能够主动向连接的客户端推送实时数据。SignalR适用于构建即时聊天、实时协作和实时更新等功能。无需复杂的轮询机制，SignalR利用WebSocket技术实现高效通信。通过简单易懂的API，使开发者能够快速集成实时特性，为应用增加动态、交互式的体验。

1. 安装SignalR

首先，在你的ASP.NET项目中，确保已经安装了SignalR库。你可以通过NuGet包管理器执行以下命令：

```
1 | Install-Package Microsoft.AspNet.SignalR
```

2. 服务端基础功能

在你的项目中，创建一个SignalR Hub类。这是一个简单的聊天室示例：

```
1 using Microsoft.AspNet.SignalR;
2
3 public class ChatHub : Hub
4 {
5     // 定义客户端调用的方法
6     public void SendMessage(string userName, string message)
7     {
8         // 将消息广播给所有连接的客户端
9         Clients.All.broadcastMessage(userName, message);
10    }
11 }
```

3. 配置SignalR

确保在你的Startup.cs文件中启用SignalR：

```
1 using Microsoft.Owin;
2 using Owin;
3
4 [assembly: OwinStartup(typeof(YourNamespace.Startup))]
5
6 namespace YourNamespace
7 {
8     public class Startup
9     {
10        public void Configuration(IAppBuilder app)
11        {
12            // 启用SignalR
13            app.MapSignalR();
14        }
15    }
16 }
```

4. 客户端基础功能

在你的客户端页面中，添加SignalR客户端脚本，并连接到Hub：

```
1 <!-- 引入SignalR客户端脚本 -->
2 <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
3 <script src="~/Scripts/jquery.signalR-2.4.2.min.js"></script>
4 <script src="~/signalr/hubs"></script> <!-- 自动生成的SignalR代理脚本 -->
5
6 <script>
7     $(function () {
8         // 连接到Hub
9         var chatHub = $.connection.chatHub;
10
11        // 客户端接收服务器发送的消息
12        chatHub.client.broadcastMessage = function (userName, message) {
13            // 处理接收到的消息
14            console.log(userName + ' says ' + message);
15        };
16
17        // 启动连接
18        $.connection.hub.start().done(function () {
19            // 可以在这里进行一些初始化操作
20        });
21
22        // 发送消息的示例
23        $('#sendMessageButton').click(function () {
24            var userName = $('#userNameInput').val();
25            var message = $('#messageInput').val();
26
27            // 调用服务器端的SendMessage方法
28            chatHub.server.sendMessage(userName, message);
29        });
30    });
31 </script>
```

5. 高级功能 - Groups

SignalR支持将连接分组，以便向特定组广播消息。例如，在Hub中添加以下方法：

```
1 public class ChatHub : Hub
2 {
3     public void JoinGroup(string groupName)
4     {
5         Groups.Add(Context.ConnectionId, groupName);
6     }
7
8     public void SendMessageToGroup(string groupName, string userName, string message)
9     {
10        Clients.Group(groupName).broadcastMessage(userName, message);
11    }
```

```
12 |     }  
    }
```

客户端可以通过调用`JoinGroup`方法加入组，然后使用`SendMessageToGroup`方法向特定组广播消息。

这是一个简单而基础的SignalR实时通信示例。在实际项目中，你可能需要更多的功能和安全性措施，例如处理连接和断开事件、用户身份验证等。确保在生产环境中使用安全的实践。