**SignalR Introduction**

**Interface for hub**

**1. With Interface**

a. Interface

A picture containing graphical user interface

Description automatically generated

b. Inherit interface to hub

Text

Description automatically generated

c. Now signalr client can direct execute the interface action

Text

Description automatically generated

2. Without interface



**Type of client/connection**

1. Clients.All – All people with receive

2. Clients.Caller – Only the sender him/her self can receive

3. Clients.Groups([groupName]) – only specific group of users can receive

4. Clients.Others – except sender, other will receive

5. Clients.GroupExcept([groupName],[new []{excludedConnectionIds}) – specific group member will receive the message except excluded connection id.

6. Clients.User([userId]) – only specific user with userId can receive (need setup something, will discuss in dept in later part)

7. Clients.Users(new []{userId\_list}) – only specific users with user id will receive

8. Clients.Client([connectionId]) – similar .User, but base on connectionId

9. Clients.Clients(new[]{connectionList}) – similar .Users, but base on connectionId

10. Clients.AllExcept(new[]{connectionList}) – all users with receive except specific connection id users.

**SignalR UserIdProvider**

Connection id will refresh for each instance, mean it’s not persistence value.

So in order to have an persistence value to send to specific user, we can let use client.user method with user id.

The user id will save in cookie or any format of authentication value. But signalr hub will not know which value is storing the user id.

So we need to override signal useridprovider to let it know which key store the user id value.

To override useridprovider:

Text

Description automatically generated

And then register it:



Don’t forget add Authorize attribute to the hub class

Graphical user interface, text

Description automatically generated

Your can get current user Id by Context.UserIdentifier in the hub class

Or

You can access claim through Context.User.Claims

**HubFilter**

HubFilter is the middleware for signalr hub. We can override this middleware before signalr invoke any methods.

In the following case, we want have the real time authentication checking before the signal action is executed.

1. Add expiry time to claim once get the authenticate.

Text

Description automatically generated

2. Abstract IHubFilter and override

Text

Description automatically generated

3. can override other hub method like onConnected or OnDisconnected etc.

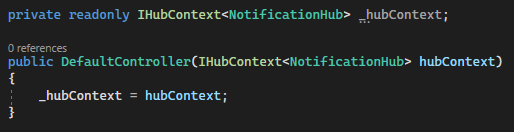
4. Register hubfilter

Text

Description automatically generated

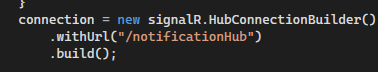
**Call SignalR from services**

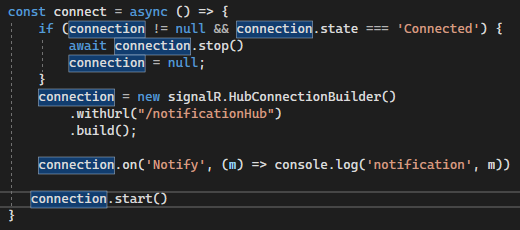




Javascript

1. Make Connection





2. Listening to event

Connection.on(‘eventName’,(data) => doSomething)

3. Stop

Connection.stop()

4. Cut unsubscribe an event

connection.off(‘event\_name’)

5. send & invoke method

Connection.send(‘event\_name’,data) // no return promise,can look as void function

Connection.invoke(‘event\_name’,data) // return promise and value(if any)