

# Chatter with SignalR

Program a chatter with Angular and .Net Core using SignalR as the main communication channel.

### Angular SignalR Chatter

Name:  Pwd:

Timestamp	Name	Message
17:38:00		Client Admin connected
17:38:18		Client Pepi connected
17:38:20	Pepi	Halli Hallo
17:38:32	Hansi	adfasdf
17:38:36		Client Pepi disconnected

### Angular SignalR Chatter

Name:  Pwd:    Admin 2 clients

Timestamp	Name	Message
17:38:18		Client Pepi connected
17:38:20	Pepi	Halli Hallo
17:38:32	Hansi	adfasdf
17:38:36		Client Pepi disconnected
17:46:15	Admin	All users requested

## ClientRepository

Create a service ClientRepository that stores Username, RegisterTime and LastMessageTime for every connected client in a dictionary (key is the ConnectionId).

This service will be used by both the hub and the controller (see below).

## Hub

Create a hub with the endpoint "/chat" in Program.cs. Use a typed hub, i.e. create an interface IChatClient with all required methods to be called on the client.

```
public interface IChatClient
{
    Task NewMessage(string name, string message, string timestamp);
    Task ClientConnected(string name);
    Task ClientDisconnected(string name);
    Task AdminNotification(string message);
    Task NrClientsChanged(int nr);
}
```

Some notifications will only be sent to the Admin group (see below).

## SignIn/SignOut

With the methods `bool SignIn(string username, string pwd)` and `void SignOut()` a client can sign in to the Chatter.

Take care of the following:

- If the length of the password is less than 5 throw an exception. This exception must be displayed on the client.
- If the username starts with "Admin" the user is treated as an administrator.
- The return value of SignIn is true, if the user is an administrator
- The user is added to/removed from the ClientRepository
- A ClientConnected/ClientDisconnected message is sent to all other clients
- If the client disconnects without explicitly calling SignOut, it will be disconnected anyway.

## Message

A user can send messages to others. These messages are distributed by the hub to all others. Don't forget to update LastMessageTime in the repository.

## Administrators

Administrators will get additional information from the backend:

- New number of clients, if a client signs in or out
- Anytime someone request all connected clients from the controller (see below).
- On the frontend an administrator has to request the current number of clients after signing in.

## Controller

Write a controller with two actions.

### AllUsers

Get some information about the connected users. Supply the timestamps in a readable format:

```
[
  {
    "name": "Admin",
    "registeredString": "17:38:00",
    "lastMessageTimeString": "00:00:00"
  },
  {
    "name": "Hansi",
    "registeredString": "17:37:44",
    "lastMessageTimeString": "17:38:32"
  }
]
```

Additionally inform all administrators, whenever someone calls a controller method.

### Broadcast

Send a message to all users via a HttpPost.

### Extension

Every user can specify a list of topics she is interested in. This will be stored in the ClientRepository. When sending a message, now additionally a topic can be added. If a topic is given, only those clients are informed, that are interested in that topic.

Name:  Pwd:

Topics of interest:

Topic:

Timestamp	Name	Message
18:35:48		Client Admin connected
18:36:15		Client Susi connected
18:36:42	Hansi	Hallo General
18:36:56	Admin	Hallo General
18:37:09	Admin	Hallo Sports

Name:  Pwd:

Topics of interest:

Topic:

Timestamp	Name	Message
18:35:48		Client Admin connected
18:36:05		Client Hansi connected
18:36:42	Hansi	Hallo General
18:36:56	Admin	Hallo General

Content:

```
[
  {
    "name": "Admin",
    "registeredString": "18:35:48",
    "lastMessageTimeString": "18:37:09",
    "topicsOfInterest": [
      "General",
      "Sports"
    ]
  },
  {
    "name": "Hansi",
    "registeredString": "18:36:05",
    "lastMessageTimeString": "18:36:42",
    "topicsOfInterest": [
      "General",
      "Sports"
    ]
  },
  {
    "name": "Susi",
    "registeredString": "18:36:15",
    "lastMessageTimeString": "00:00:00",
    "topicsOfInterest": [
      "General"
    ]
  }
]
```