NEW MEDIA & COMMUNICATION TECHNOLOGY

ASP.NET MVC Labo Week 11

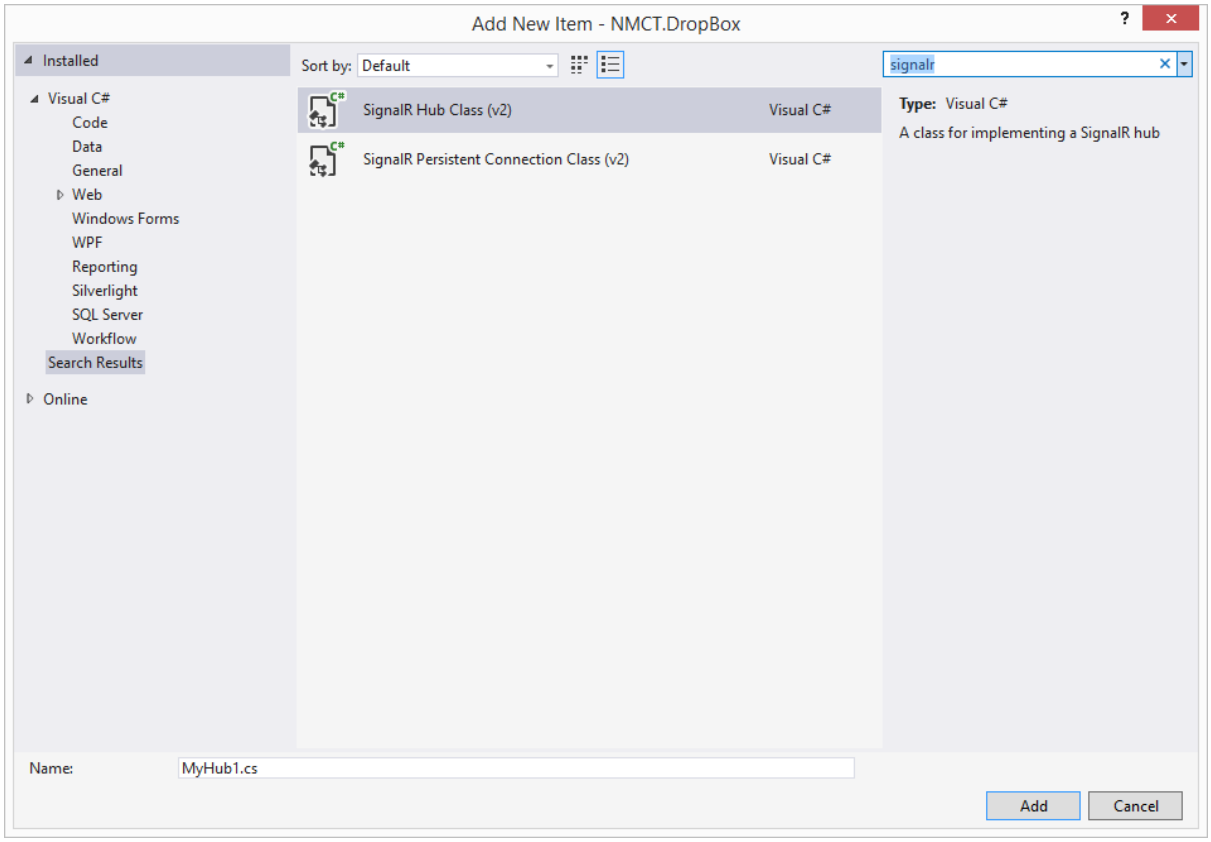
Doelstelling:

- Integratie van SignalR in dropbox applicatie
- Afwerken dropbox oefening

Inbouwen van chat functionaliteit in de dropbox appliatie zodat we altijd zien hoeveel files er tot nu toe gedownload zijn en hoeveel files er reeds geupload zijn, en dit zonder page refresh.

Aanmaken Counter Hub

Maak een folder “Hubs” aan binnen je dropbox applicatie. In deze folder voegen we een SignalR hub toe via “Add New Item” en we noemen deze Counters



Zorg ervoor dat deze klasse erft van Hub:

```
1 reference
public class Counters : Hub
{

}
```

Open nu Startup.Auth.cs en voeg de SignalR mapping bovenaan toe:

```
public void ConfigureAuth(IAppBuilder app)
{
    app.MapSignalR(new HubConfiguration() { EnableDetailedErrors = true });
```

Configureren van de Javascript Client

Open de pagina waar je de files toont. Voeg bovenaan de scripts naar SignalR toe. Het eerste script is een verwijzing naar de client side signalR library. Het tweede script zal dynamisch gegenereerd worden bij het opstarten van de applicatie.

```
@model IEnumerable<NMCT.DropBox.Models.FileRegistration>
<script src="~/Scripts/jquery.signalR-2.1.2.min.js"></script>
<script src="~/signalr/hubs"></script>
```

Voeg hieronder script tags toe waarin we later de javascript code voor communicatie met de server kunnen schrijven. Declareer ook de Jquery load function.

```
<script>

    $(function () {

    })

</script>
```

Controleer in _Layout.cshtml of je bovenaan de Jquery script niet vergeet in te laden:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
```

Downloaden Files

Het eerste wat we wensen te doen is weergaven hoeveel files er reeds werden gedownload en dit in realtime weergeven. Dus telkens iemand een file download moeten we de teller in realtime zien verhogen. Dus GEEN page refresh.

Ga nu naar de client code en wijzig de JQuery load function. Eerst gaan we een referentie maken naar de counters hub via $.connection.counters. Daarna moeten we van zodra de connectie met de hub ok is het aantal gedownloade files ophalen vanuit de server. We moeten dus de server methode GetDownloadedFiles() aanroepen. We doen dit door via de referentie counters de server property aan te spreken gevolgd door getDownloadedFiles().

```
$(function () {

    var counters = $.connection.counters;

    $.connection.hub.start().done(function () {
        counters.server.getDownloadedFiles();
    });
```

Nu moeten we nog deze server methode schrijven. Ga naar de Counters hub en voeg volgende methode toe. In deze methode moet u naar de database gaan en ophalen hoeveel files er reeds werden gedownload. U moet zelf instaat zijn om dit te schrijven. Het aantal files moet u dan doorgeven aan de methode Clients.All.NumberOfFilesDownloaded(…). Deze methode zal op de CLIENT aangeroepen worden en zal het resultaat naar ALLE clients doorsturen.

```
0 references
public void GetDownloadedFiles()
{
    int aantal = "<< ga hier naar de database access klasse en haal het totaal aantal gedownloade files op >>";
    Clients.All.NumberOfFilesDowloaded(aantal);
}
```

Aan de client kant moeten we deze methode opvangen en het resultaat weergeven. De client code ziet er als volgt uit. Vanaf de server roepen we deze methode aan met het resultaat. Via

```
counters.client.numberOfFilesDowloaded = function (numberOfFiles) {
    $('#downloadedFiles').text("Totaal aantal bestanden gedownload:" + numberOfFiles);
}
```

Voeg onderaan volgende HTML code toe om het resultaat weer te geven:

```
<p id="downloadedFiles"></p>
```

Test uit en kijk of het resultaat verschijnt in de pagina.

Nu moeten we er nog voor zorgen dat bij iedere download het aantal op de client een nieuwe update ontvangt. Het is mogelijk om vanuit een ASP.NET controller SingalR hubs aan te spreken. Onderstaande code zal dit doen voor u: merk op, u moet zelf nog het aantal files ophalen die werd gedownload.

```
var hub = Microsoft.AspNet.SignalR.GlobalHost.ConnectionManager.GetHubContext<Counters>();
hub.Clients.All.NumberOfFilesDowloaded(numberOfLogFlies);
```

Test uit met verschillende gebruikers. Vanaf nu zou bij iedere download het aantal moeten verhogen in de browser.

Uploaden files

U zou nu zelf instaat moeten om weer te geven hoeveel files er reeds geupload werden en bij iedere nieuwe upload dit in realtime weer te geven.