NEW MEDIA & COMMUNICATION TECHNOLOGY

ASP.NET MVC Labo Week 5

Doelstelling:

- Introductie tot ADO.NET
- Herhaling ASP.NET MVC Concepten

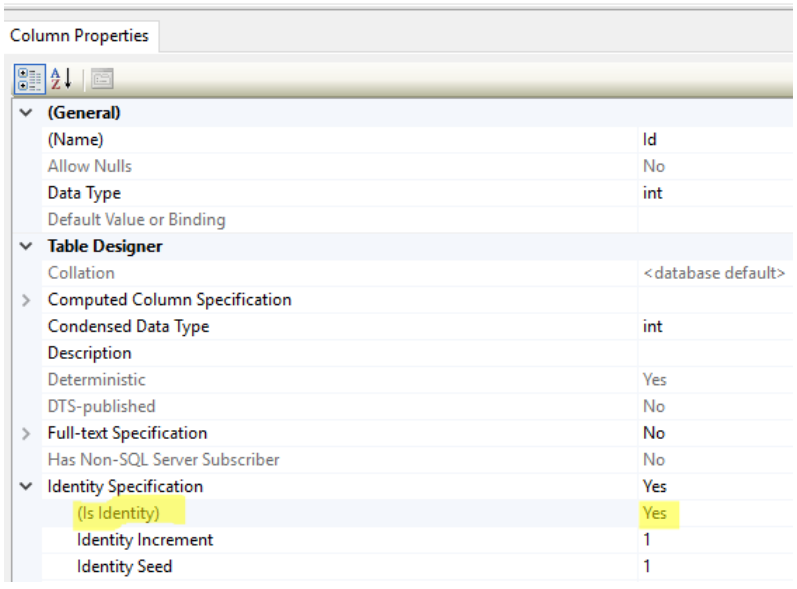
Oefening 1: Events (Samen)

We maken gebruik van de oefening over events en sessions van de laatste weken. Tot nu maakten we gebruik van Data.cs waarin reeds alle data aanwezig was. Nu gaan we ervoor zorgen dat de data uit een SQL Server zal ingeladen worden.

Stap 1: SQL Server + Database aanmaken

Maak een nieuwe database aan binnen SQL Server met als naam “ASPMVCSessieOefening”. In deze database maken we een tabel “Sessies” met volgende velden:

Id	int	☐
Name	nvarchar(50)	☑
Slot	int	☑
Description	nvarchar(1000)	☑
		☐



Deze velden komen overeen met de properties in ons model “Sessie”. Wat is er speciaal aan het veld “Id” ?. Meer info over de datatype mappings kan u vinden op https://msdn.microsoft.com/en-us/library/cc716729(v=vs.110).aspx

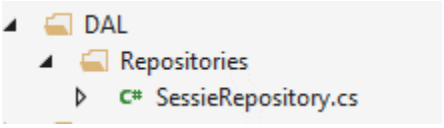
```
public class Sessie
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int Slot { get; set; }
    public string Description { get; set; }
}
```

Stap 2: Verbinding database met applicatie

Onze applicatie zal communiceren via de connectionstring met de database. Wij maken gebruik van Windows Authentication. Zoek de juiste connectionstring op http://www.connectionstrings.com/

Stap 3: Structuur DAL

Maak een map DAL aan met daarin de map Repositories. Per tabel gaan we een repository aanmaken. We starten met de SessieRepository.



Stap 4: SessieRepository

Deze klasse zal alle code bevatten om data op te vragen, weg te schrijven, te wijzigen of te verwijderen (CRUD operaties).

AddSessie

```
public int AddSessie(Sessie sessie) {
    using (SqlConnection con = new SqlConnection(CONNECTIONSTRING)) {
        con.Open();
        using (SqlCommand command = new SqlCommand()) {
            string sql = "INSERT INTO Sessies Values (@Name,@Slot,@Description);select @@IDENTITY";
            command.Parameters.Add(new SqlParameter("@Name", sessie.Name));
            command.Parameters.Add(new SqlParameter("@Slot", sessie.Slot));
            command.Parameters.Add(new SqlParameter("@Description", sessie.Description));
            command.CommandText = sql;
            command.Connection = con;
            int id = int.Parse(command.ExecuteScalar().ToString());
            return id;
        }
    }
}
```

NumerOfSessies

```
public int NumberOfSessions() {
    using (SqlConnection con = new SqlConnection(CONNECTIONSTRING)) {
        con.Open();
        using (SqlCommand command = new SqlCommand()) {
            string sql = "SELECT Count(*) FROM Sessies";

            command.CommandText = sql;
            command.Connection = con;
            int id = (int)command.ExecuteScalar();
            return id;
        }
    }
}
```

GetSessies()

```
public List<Sessie> GetSessies() {
    List<Sessie> sessies = new List<Sessie>();
    using (SqlConnection con = new SqlConnection(CONNECTIONSTRING)) {
        con.Open();
        using (SqlCommand command = new SqlCommand()) {
            string sql = "SELECT * FROM Sessies";
            command.CommandText = sql;
            command.Connection = con;
            using (SqlDataReader reader = command.ExecuteReader()) {
                while (reader.Read()) {
                    Sessie sessie = new Sessie();
                    sessie.Id = int.Parse(reader["Id"].ToString());
                    sessie.Name = reader["Name"].ToString();
                    sessie.Slot = int.Parse(reader["Slot"].ToString());
                    sessie.Description = reader["Description"].ToString();
                    sessies.Add(sessie);
                }
            }
        }
    }
    return sessies;
}
```

GetSessies(int slot)

Zelfstandig

GetSessie(int id)

Zelfstandig

Stap 5: Vullen database

We wensen niet alle data uit Data.cs over te tikken. Daarom gaan we een “SetupController” maken. In de methode index gaan we kijken of er al iets in de tabel sessies zit. Indien niet dan halen we deze op en voegen we deze toe aan de SQL Server. U moet dit kunnen doen op basis van de methodes uit stap 4.

Doet dit nu ook voor “Organizations”. Maak tabellen aan in deze database en zorg ervoor dat de “SetupController” deze zal invullen.

Stap 6: Registratie

Maak zelf een tabel aan waarin we registraties wensen op te slaan. Voorzie de nodig relaties in de database

Oefening 2: Verder werken op theorie voorbeeld

In deze oefening moet u de demo uit de theorie gaan uitbreiden. U kan de demo + database downloaden van LEHO. Maak een scherm waar je volgende zaken kan wijzigen:

- Zijn naam
- Zijn voornaam
- De stad waaraan deze is gekoppeld

Daarnaast moet u ook een scherm bouwen waar je een “Person” kan wijzigen.

Oefening 3: Zoekertjes

Wijzig de oefening van de zoekertjes zodat deze volledig met een database zal werken.