

ASP.NET MVC Labo

Doelstelling:


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


Oefening 1: Events Database

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 “Events”. In deze database maken we een tabel “Sessies” met volgende velden:

	Id	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input checked="" type="checkbox"/>
	Slot	int	<input checked="" type="checkbox"/>
	Description	nvarchar(1000)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Column Properties	
	
▼ (General)	
(Name)	Id
Allow Nulls	No
Data Type	int
Default Value or Binding	
▼ Table Designer	
Collation	<database default>
> Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
> Full-text Specification	No
Has Non-SQL Server Subscriber	No
▼ Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

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](https://msdn.microsoft.com/en-us/library/cc716729(v=vs.110).aspx)

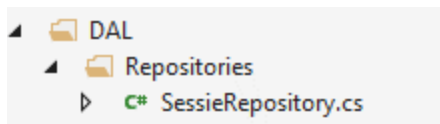
```
public class Sessie
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int Slot { get; set; }
    public string Description { get; set; }
}
```

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/>. Zorg ervoor dat je de settings (connectionstring en provider) goed inleest. Hoe ging dit weer? Bekijk de les demo indien nodig. Vergeet niet dat we dependency injection gebruiken.

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).

- GetSessies()
- GetSessies(int slot)
- GetSessie(int id)

Stap 5: Vullen database

Vul nu de database met sessies via de SQL Enterprise Manager. Schrijf dus manueel de SQL Statement zodat er sessies in de database zitten.

Stap 6: Wijzigen applicatie

Wijzig nu uw de applicatie zodat we gebruik maken van de repository. Maw. Vanuit de controllers gaan we nu de SessieRepository gebruikt en niet de Data.cs. Doet dit zowel voor gewone controllers als API. Indien nodig legt u de nodige relaties tussen de tabellen.

Stap 7: Andere tabellen

Het is de bedoeling dat we volledig zonder Data.cs eindigen. Maak dus de rest van de tabellen aan. Vul deze met de data uit Data.cs. Voeg de nodige repositories toe. Binnen de nodig repositories

moet u de nodig methodes toevoegen om te praten met de database. Wijzig ook de controllers. Uiteindelijk moet je de file Data.cs kunnen wissen. Dan pas werkt alles met een database.

Stap 6: Registratie

Maak zelf een tabel aan waarin we registraties wensen op te slaan. Voorzie de nodig relaties in de database. Voorzie een registratie repository. Schrijf daar de code voor de SQL INSERT.