

Oefening Music Store – deel 1

In deze oefening programmeren we een webshop voor muziekalbums met als naam *Music Store*. Via de webshop kan je albums opzoeken, in je winkelkarretje stoppen en vervolgens naar de kassa om af te rekenen. We programmeren zowel het gebruikersgedeelte als het administrator-gedeelte.

Tijdens dit eerste deel maken we de website aan, creëren we de tabellen en voorzien we de beveiliging van de site.

Maak in de map *www* een nieuwe *ASP.NET Core Webapplication* met als naam *MusicStore*. Gebruik als *Authentication Individual User Accounts*.

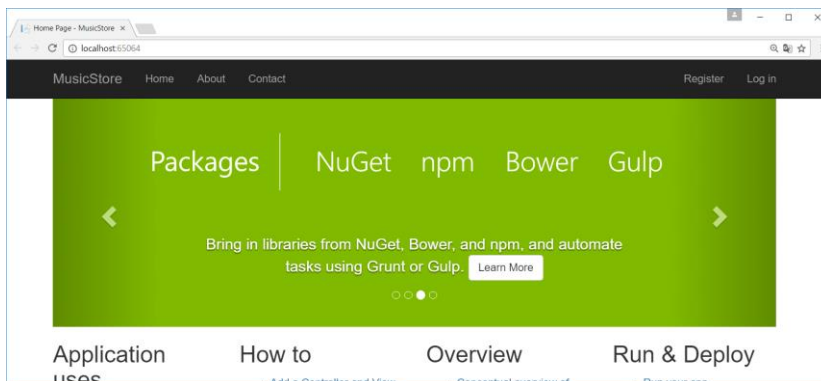
Voor het ontwikkelen van de website heb je een aantal bestanden nodig. Kopieer de inhoud van de mappen *css* en *images* onder *wwwroot* naar de nieuwe applicatie. Bestaande bestanden mag je overschrijven. Kopieer ook de volledige map *content* naar *wwwroot* van de applicatie. In deze map vind je de meeste covers van albums die verkocht worden in de shop.

1. Models

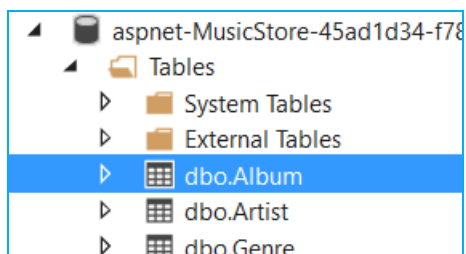
Kopieer in de map *Models* de drie klassen voor onze webshop: *Album*, *Artist* en *Genre*. Doe de klassen open en bekijk de inhoud zodat de structuur duidelijk is.

Kopieer de klassen *StoreContext.cs* en *DBInitializer.cs* naar de *Data-map*. Bekijk ook deze klassen aandachtig!

Doe nu zelf de laatste aanpassingen aan je project zodat bij het opstarten de database en de tabellen gecreëerd worden en de data wordt toegevoegd. Je krijgt voorlopig:



Open de *SQL Server Object Explorer* om te controleren of de tabellen er zijn:



De tabel *Album* bijvoorbeeld:

| dbo.Album [Data] | | | | | | |
|---|---------|----------------|----------|---------|-------|-----------------|
| Startup.cs DbInitializer.cs StoreContext.cs | | | | | | |
| Max Rows: 1000 | | | | | | |
| | AlbumID | AlbumArtUrl | ArtistID | GenreID | Price | Title |
| 1 | | cover0001.p... | 1 | 9 | 9 | The Best Of ... |
| 2 | | cover0157.p... | 18 | 5 | 9 | The Essentia... |
| 3 | | placeholder... | 17 | 3 | 9 | Milton Nasc... |
| 4 | | cover0159.p... | 17 | 3 | 9 | Minas |
| 5 | | placeholder... | 8 | 4 | 9 | Motley Crue... |
| 6 | | placeholder... | 2 | 4 | 9 | Ace Of Spa... |
| 7 | | placeholder... | 3 | 7 | 9 | Mozart: Cha... |
| 8 | | cover0163.p... | 4 | 7 | 9 | The Best of ... |
| 9 | | cover0164.p... | 5 | 9 | 9 | Nevermind |
| 10 | | placeholder... | 6 | 9 | 9 | Compositor... |
| 11 | | placeholder... | 7 | 3 | 9 | Olodum |

2. Security

Een deel van onze webshop mag alleen toegankelijk zijn voor een *Administrator*. Zorg ervoor dat het beveiligingssysteem van *ASP.NET MVC* geactiveerd wordt.

Pas eerst de migraties toe zodat de tabellen gecreëerd worden:

| Tables | |
|--------|--------------------------|
| ▷ | System Tables |
| ▷ | External Tables |
| ▷ | dbo._EFMigrationsHistory |
| ▷ | dbo.Album |
| ▷ | dbo.Artist |
| ▷ | dbo.AspNetRoleClaims |
| ▷ | dbo.AspNetRoles |
| ▷ | dbo.AspNetUserClaims |
| ▷ | dbo.AspNetUserLogins |
| ▷ | dbo.AspNetUserRoles |
| ▷ | dbo.AspNetUsers |
| ▷ | dbo.AspNetUserTokens |
| ▷ | dbo.Genre |

Zorg er dan voor dat de beperkingen voor wachtwoorden niet zo sterk zijn. Je kan uiteraard code kopiëren uit je *Contoso University* project.

Maak daarna een *administrator* admin@musicstore.com met wachtwoord *musicstore* aan. Creëer een *Administrator Role* en voeg de admin-gebruiker eraan toe.

3. Admin Area

Maak een aparte *Area* voor alle *Admin*-functionaliteiten. Stop in die *Area* een controller *AlbumsController.cs* met één methode *Index*. Deze methode roept een *view* op. Maak die *view* als volgt aan:

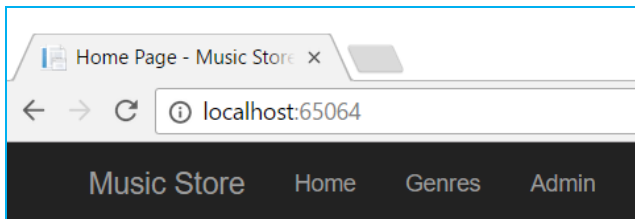
```
@{
    ViewData["Title"] = "Albums Index";
}

<h2>@ViewData["Title"]</h2>

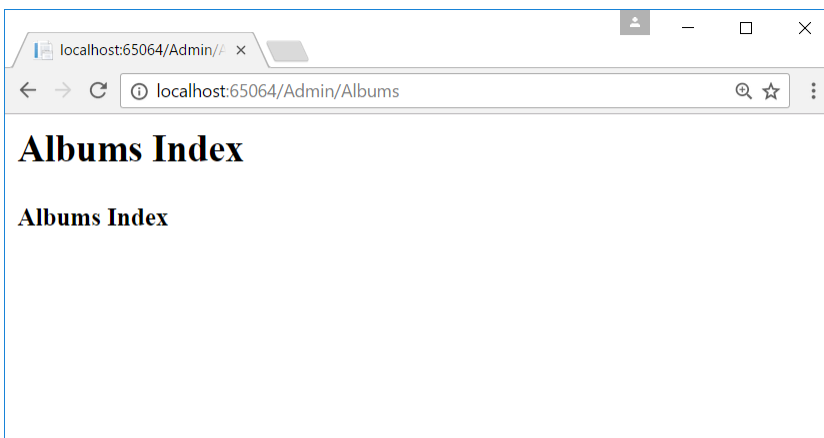
<h4>Albums Index</h4>
```

Doe de juiste aanpassingen voor het gebruik van *Areas* aan *Startup.cs* en *Project.json*.

Pas de *layout*-template van de site aan en schrijf op drie plaatsen *Music Store* in twee woorden. Doe *About* en *Contact* in het menu weg. Maak een menu-item *Genres* waarmee je springt naar de *ListGenres*-methode van de *Store*-controller. Voorzie ook een menu-item *Admin* waarmee je springt naar de *Index*-methode van de *Albums*-controller (in de *Admin Area*) van daarnet:



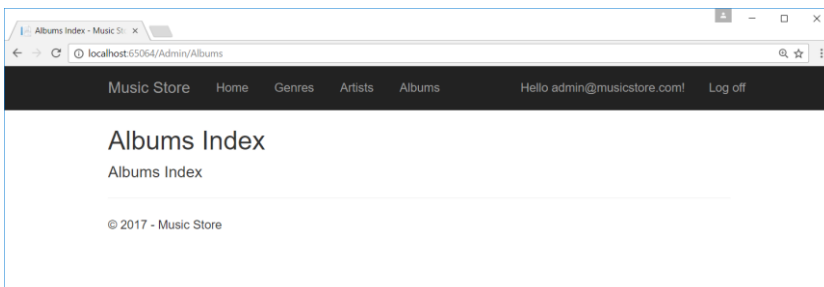
Controleer of je via het menu-item *Admin* de *Albums Index* pagina kan oproepen:



4. Admin layout

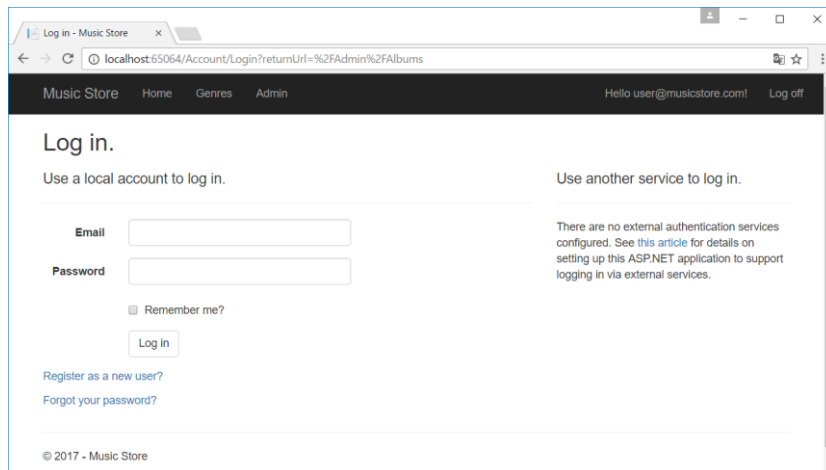
Zorg ervoor dat voor de *Admin* pagina's een andere *layout*-pagina *_LayoutAdmin.cshtml* wordt gebruikt met bovenaan de menu-items *Genres*, *Artists* en *Albums*. Je roept telkens de *Index*-methode op (de controllers bestaan trouwens nog niet, dat komt later). Al deze controllers komen later in de *Admin*-Area. Doe de juiste *Default Imports* voor de *tag helpers*, ...

Probeer uit of de *Albums Index* pagina de juiste opmaak en menu heeft:



5. Authorization

Zorg ervoor dat alleen de gebruikers van de *Administrator Role* aan de *Albums*-pagina kunnen. Maak een tweede gebruiker aan en controleer of die al dan niet aan de pagina kan. Gebruikers die niet genoeg rechten hebben, worden omgeleid naar de *Login* pagina:



6. Home pagina

We zorgen tenslotte voor een aangepaste *homepage*. De code ervan kan je gewoon uit de resources kopiëren. Je krijgt:

