

2018

Object georiënteerd  
programmeren met C# en  
ASP.NET/MVC incl. database



ROC van Twente

16-7-2018



---

## Inhoud

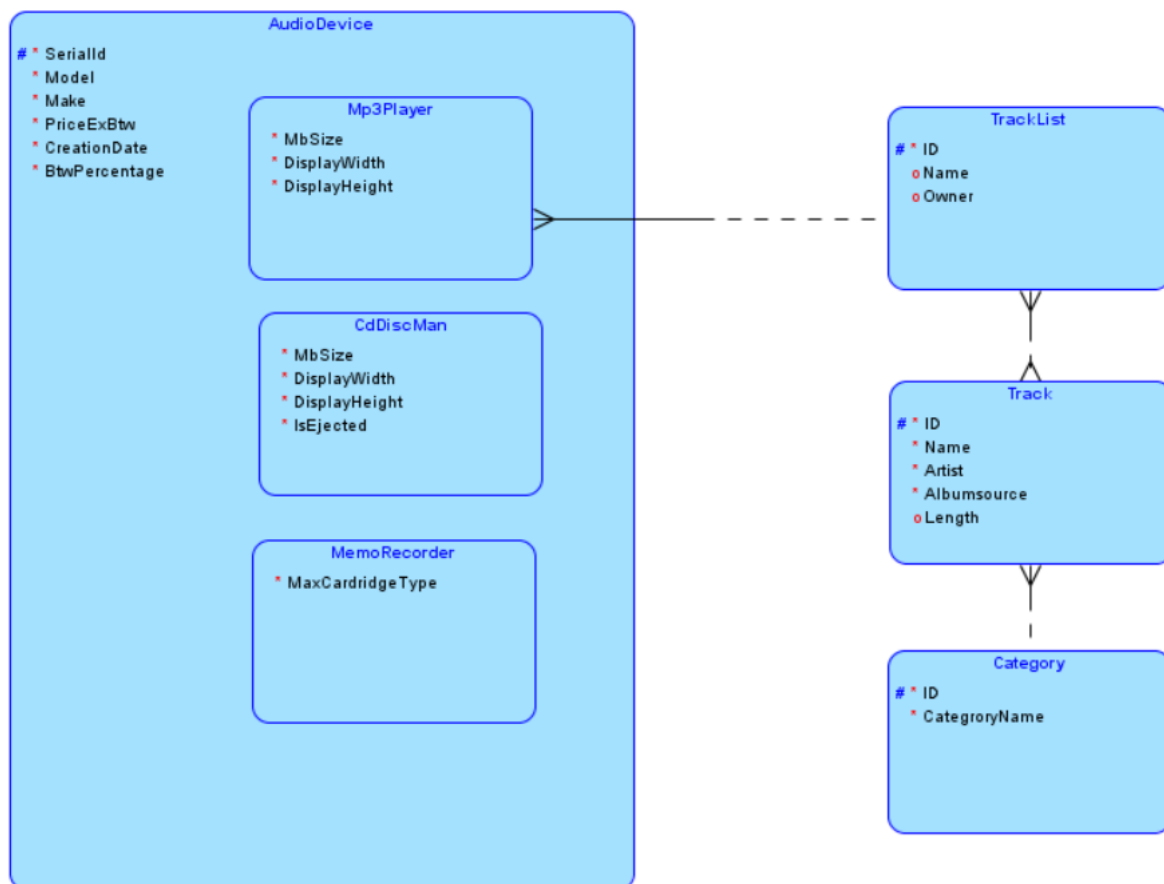
Het database ontwerp.....	3
Opdracht 1 - De <i>Track</i> tabel en de tabel <i>Category</i> .....	4
Opdracht 2 - De <i>TrackList</i> tabel .....	5
Opdracht 3 - De <i>AudioDevice</i> tabel .....	6
Opdracht 4 – De tabellen <i>MemoRecorder</i> , <i>CdDiscman</i> en <i>Mp3Player</i> .....	7
Opdracht 5 - De <i>TrackList</i> tabel koppelen aan <i>Mp3Player</i> .....	9
Opdracht 6 – Voorbeelddata aan de tabellen toevoegen.....	10
Stap 1: Categorieën toevoegen .....	10
Stap 2: Tracks toevoegen .....	10
Stap 2: Audiodevices toevoegen .....	10
Stap 2a: Memorecorders.....	10
Stap 2b: CdDiscmans .....	11
Opdracht 7 – MVC website maken m.b.v. Visual Studio.....	11
Stap 1: Nieuw project aanmaken .....	11
Stap 2: Class Library aan solution toevoegen .....	12
Stap 3: Data Model toevoegen aan class library .....	12
Stap 4: Enkele huishoudelijke taken in de solution.....	18
Stap 5: Categorieën beheren.....	20
Stap 6: Tracks beheren .....	23
Stap 7: De menubalk aanpassen .....	24
Stap 8: Beheertaken alleen voor beheerders.....	24
Stap 9: Devices beheren .....	27
Stap 9a: Een ViewModel voor een MemoRecorder .....	27
Stap 9d: Een submenu toevoegen aan de menubalk voor audiodevices .....	33
Stap 10: Tracks aan een tracklist toevoegen.....	34



## Het database ontwerp

SoundSharp wil de eerder gemaakte applicatie gaan vervangen door een versie die gebruik maakt van een database. Informatie over tracks en geluidsdragers moet nu worden opgeslagen in een SQL Server database.

Het ontwerpen van de database is uitbesteed aan een HBO-student als stageopdracht. Zij kwam met het volgende Entity Relationship Model op de proppen (gemaakt met behulp van de Oracle Data Modeler, welke gratis gedownload kan worden van de [site van Oracle](#)):



Enige toelichting:


- **Mp3Player**, **MemoRecorder** en **CdDiscman** zijn subtypes van **AudioDevice**.
- Een **TrackList** KAN bij één of meerdere Mp3Players horen.
- Een **Mp3Player** MOET één TrackList hebben.
- Tussen TrackList en Track bestaat een veel-op-veel relatie: een **Track** KAN op één of meerdere **TrackLists** voorkomen en een **TrackList** MOET één of meerdere **Tracks** bevatten.
- Een **Category** KAN bij een of meerdere **Tracks** horen en een **Track** MOET exact één **Category** hebben.

Het gegeven ERD gaan we uitwerken in een SQL Server database




## Opdracht 1 - De *Track* tabel en de tabel *Category*

De eerste tabel die ontwikkeld wordt is de **Track** tabel. Deze tabel bevat alle MP3 tracks. Voor iedere track moet ook een genre worden gekozen en daarom wordt begonnen met het aanmaken van een tabel **Category**.

Category			
	Column Name	Data Type	Allow Nulls
	ID	int	<input type="checkbox"/>
	CategoryName	nvarchar(25)	<input type="checkbox"/>
			<input type="checkbox"/>

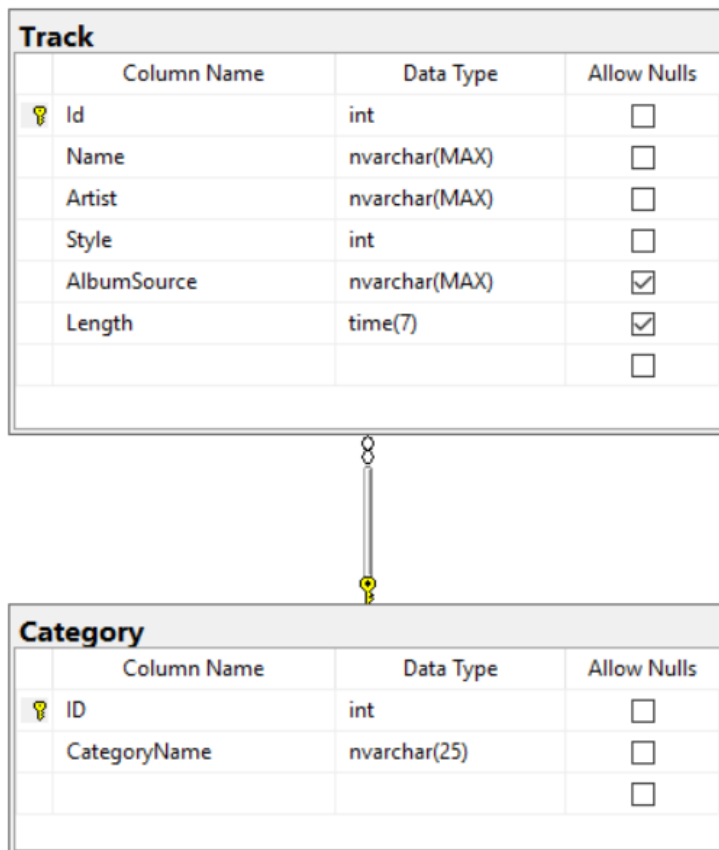
Het veld ID is de primary key. Zorg dat SQL Server automatisch een nummer toekent als een nieuwe categorie wordt toegevoegd.

Zodra we de tabel **Category** hebben aangemaakt kunnen we de tabel **Track** toevoegen.

Track			
	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	Artist	nvarchar(MAX)	<input type="checkbox"/>
	Style	int	<input type="checkbox"/>
	AlbumSource	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Length	time(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Het veld *Style* dient als foreign key naar de tabel **Category**. Deze relatie kunnen we via Sql Server Management toevoegen door een diagram toe te voegen aan de database. Vervolgens voegen we beide tabellen toe en koppelen het veld *Style* aan het veld *ID* van de tabel **Category**.

De *ID* van een track moet weer automatisch worden gegenereerd.



We kunnen de relatie lezen als:

- Een **Category** heeft één of meerdere **Tracks**
- Een **Track** heeft één **Category**. Een track moet altijd een category hebben aangezien *Style* geen NULL waarde mag bevatten.

## Opdracht 2 - De *TrackList* tabel

We vervolgen met het aanmaken van de tabel **TrackList**. Een tracklist kan natuurlijk, de naam zegt het al, meerdere tracks bevatten, maar een track kan ook bij meerdere tracklists horen. Er bestaat dus een veel-op-veel relatie. In een relationele database, zoals SQL Server, kunnen we geen veel-op-veel relaties definiëren en dat lossen we op door een koppeltabel te maken: **TrackListTrack**. We krijgen dus de volgende 2 tabellen:

TrackList			
	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	Name	nvarchar(100)	<input checked="" type="checkbox"/>
	Owner	nvarchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

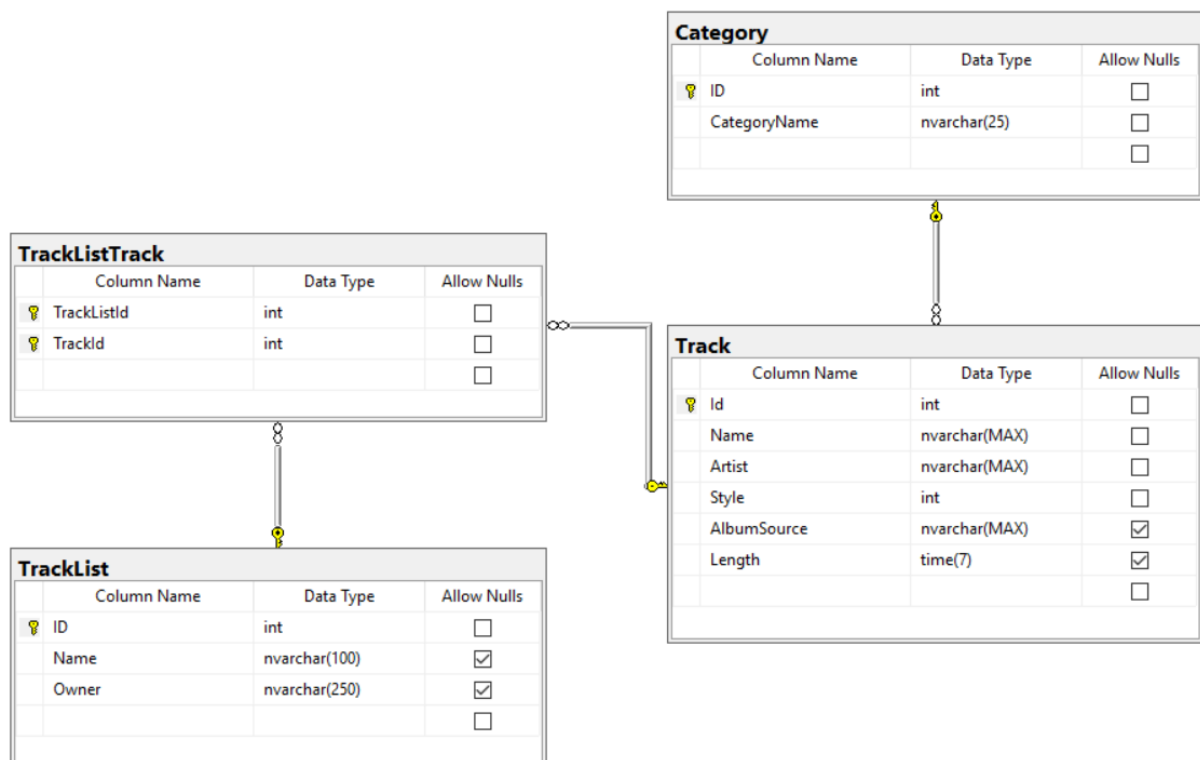


ID moet ook hier weer automatisch worden gegenereerd.

TrackListTrack			
	Column Name	Data Type	Allow Nulls
🔑	TrackListId	int	<input type="checkbox"/>
🔑	TrackId	int	<input type="checkbox"/>
			<input type="checkbox"/>

De tabel **TrackListTrack** heeft een samengestelde primary key. Hetzelfde nummer kan dus niet 2 keer op dezelfde tracklist voorkomen.

De relaties kunnen we definiëren door de tabellen toe te voegen aan het diagram.



### Opdracht 3 - De *AudioDevice* tabel

We gaan onze database vervolgens uitbreiden met een tabel **AudioDevice**.



AudioDevice			
	Column Name	Data Type	Allow Nulls
🔑	SerialId	int	<input type="checkbox"/>
	Model	nvarchar(100)	<input type="checkbox"/>
	Make	nvarchar(100)	<input type="checkbox"/>
	PriceExBtw	smallmoney	<input type="checkbox"/>
	CreationDate	smalldatetime	<input type="checkbox"/>
	BtwPercentage	numeric(3, 1)	<input type="checkbox"/>
			<input type="checkbox"/>

In dit geval wordt de primary key niet automatisch gegenereerd.

#### Opdracht 4 – De tabellen *MemoRecorder*, *CdDiscman* en *Mp3Player*

Een audiodevice is of een CdDiscMan, of een Memorecorder of een Mp3Player. Voor deze 3 soorten geluidsdragers maken we afzonderlijk tabellen aan met telkens een relatie naar het bijbehorende audiodevice.

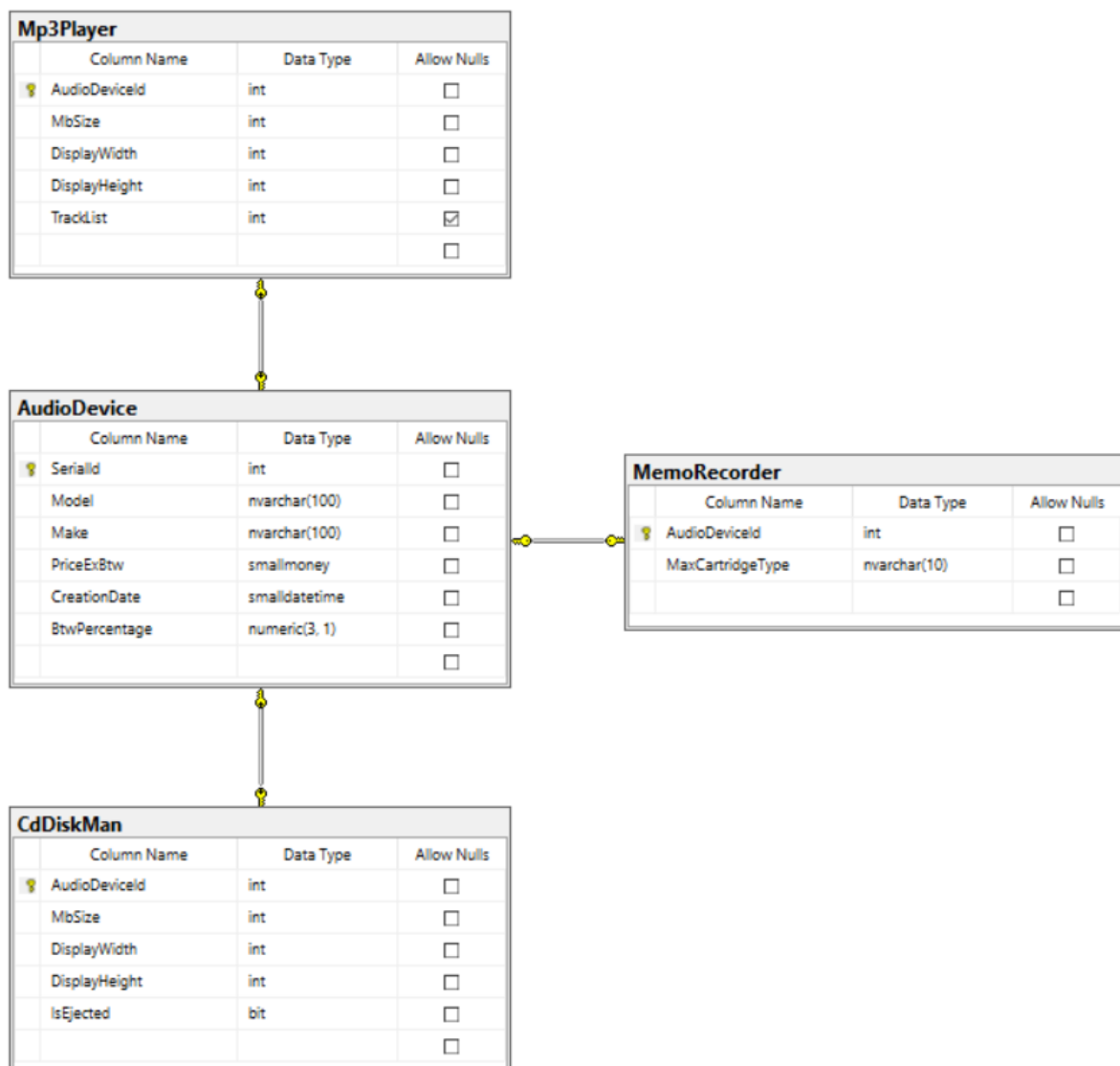
MemoRecorder			
	Column Name	Data Type	Allow Nulls
🔑	AudioDeviceId	int	<input type="checkbox"/>
	MaxCartridgeType	nvarchar(10)	<input type="checkbox"/>
			<input type="checkbox"/>

CdDiskMan			
	Column Name	Data Type	Allow Nulls
🔑	AudioDeviceId	int	<input type="checkbox"/>
	MbSize	int	<input type="checkbox"/>
	DisplayWidth	int	<input type="checkbox"/>
	DisplayHeight	int	<input type="checkbox"/>
	IsEjected	bit	<input type="checkbox"/>
			<input type="checkbox"/>



Mp3Player			
	Column Name	Data Type	Allow Nulls
🔑	AudioDeviceId	int	<input type="checkbox"/>
	MbSize	int	<input type="checkbox"/>
	DisplayWidth	int	<input type="checkbox"/>
	DisplayHeight	int	<input type="checkbox"/>
	TrackList	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

De relaties voegen we weer toe in een diagram. We kunnen voor de hele database één diagram maken, maar om het overzichtelijk te houden kunnen we ook meerdere diagrammen maken.



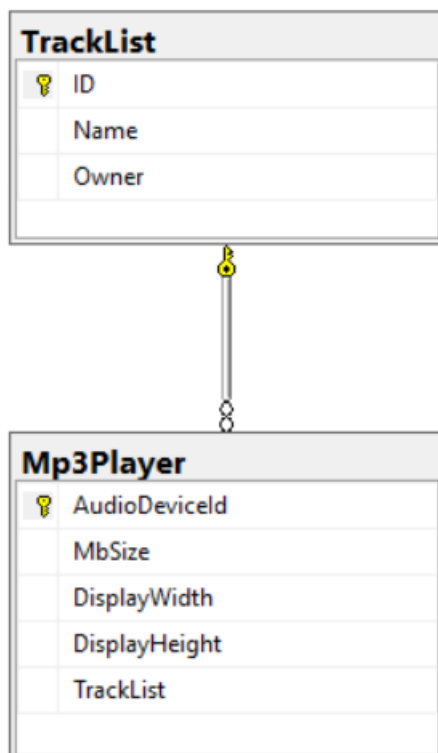




Doordat in alle 3 de geluidsdragers het serienummer van het audiodevice de primary key is 'begrijpt' SQL Server dat er één-op-één relaties gemaakt moeten worden.

### Opdracht 5 - De *TrackList* tabel koppelen aan *Mp3Player*

We hebben nu alle tabellen klaar en kunnen deze gaan vullen met wat voorbeelddata. We moeten echter nog één relatie definiëren en wel tussen de tabellen **Mp3Player** en **TrackList**. Dit doem we weer middels een database diagram:





---

## Opdracht 6 – Voorbeelddata aan de tabellen toevoegen

### Stap 1: Categorieën toevoegen

Voeg de volgende categorieën toe aan de tabel:

- Ambient
- Blues
- Country
- Disco
- Electro
- Hardcore
- HardRock
- HeavyMetal
- Hiphop
- Jazz
- Jumpstyle
- Klassiek
- Latin
- Other
- Pop
- Punk
- Reggae
- Rock
- Soul
- Trance
- Techno

### Stap 2: Tracks toevoegen

Vul de tabel **Track** met de volgende nummers:

Name	Artist	Style	AlbumSource	Length
Yellow	Coldplay	15	Parachutes	00:04:29.
Shiver	Coldplay	15	Parachutes	00:04:59.
Maneater	Nelly Futado	15	Loose	00:04:41.
Guitar	Prince	15	NULL	00:04:12.

### Stap 2: Audiodevices toevoegen

Stap 2a: Memorecorders

SerialId	Model	Make	PriceExBtw	CreationDate	BtwPercentage	MaxCartridgeT...
1	ICD-BX700	Sony	43,9900	2010-10-01 00:0...	21,0	C90
2	Voice Tracer LF	Philips	139,0000	2010-01-10 00:0...	21,0	C120
3	VN 500	Olympus	30,0000	2010-01-05 00:0...	21,0	C60



## Stap 2b: CdDiscmans

SerialId	Model	Make	PriceExBtw	CreationDate	BtwPercentage	MbSize	DisplayWidth	DisplayHeight	IsEjected
4	CD9220	Soundmaster	48,0000	2018-05-08 00:00:00	21,0	700	640	480	False
5	DM-24	Denver	24,9900	2015-12-12 00:00:00	21,0	700	640	480	False
6	CDP 6600 Pink	Grundig	52,0000	2018-10-10 00:00:00	21,0	700	1080	640	False
7	C60HF	Sony	10,9900	2011-10-09 00:00:00	21,0	700	1080	640	False

## Stap 2c: Mp3Players

SerialId	Model	Make	PriceExBtw	CreationDate	BtwPercentage	MbSize	DisplayWidth	DisplayHeight	TrackList
8	iPod Touch 6	Apple	219,0000	2018-03-04 00:00:00	21,0	32000	640	480	NULL
9	Clip Sports 8GB	Sandisk	59,9900	2017-09-09 00:00:00	21,0	8000	640	480	NULL
10	Plenue D Goud	Cowon	284,0000	2019-02-04 00:00:00	21,0	32000	1080	640	NULL
11	NW-ZX300	Sony	683,0000	2019-01-10 00:00:00	21,0	64000	1080	640	NULL

## Opdracht 7 – MVC website maken m.b.v. Visual Studio

### Stap 1: Nieuw project aanmaken

Maak in Visual Studio een nieuw ASP.NET MVC project aan met als naam **SoundsharpMVCWithDB**. Kies voor Authenticatie met Individuele User Accounts.

**New Project**

Visual C#

- Get Started
- Windows Universal
- Windows Desktop
- Web**
- Office/SharePoint
- .NET Core
- .NET Standard
- Android
- Apple TV
- Apple Watch
- Bootstrap Bundle
- Cloud
- Cross-Platform
- Documentation
- Extensibility
- iOS Extensions
- iPhone & iPad
- Test
- WCF

Sort by: Default

Search (Ctrl+E)

Icon	Project Name	Type
ASP.NET Core Web Application icon	ASP.NET Core Web Application	Visual C#
ASP.NET Web Application (.NET Framework) icon	ASP.NET Web Application (.NET Framework)	Visual C#

**Type:** Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Not finding what you are looking for?  
[Open Visual Studio Installer](#)

Name: SoundSharpUI

Location: C:\Users\Gebruiker\source\repos

Solution: Create new solution

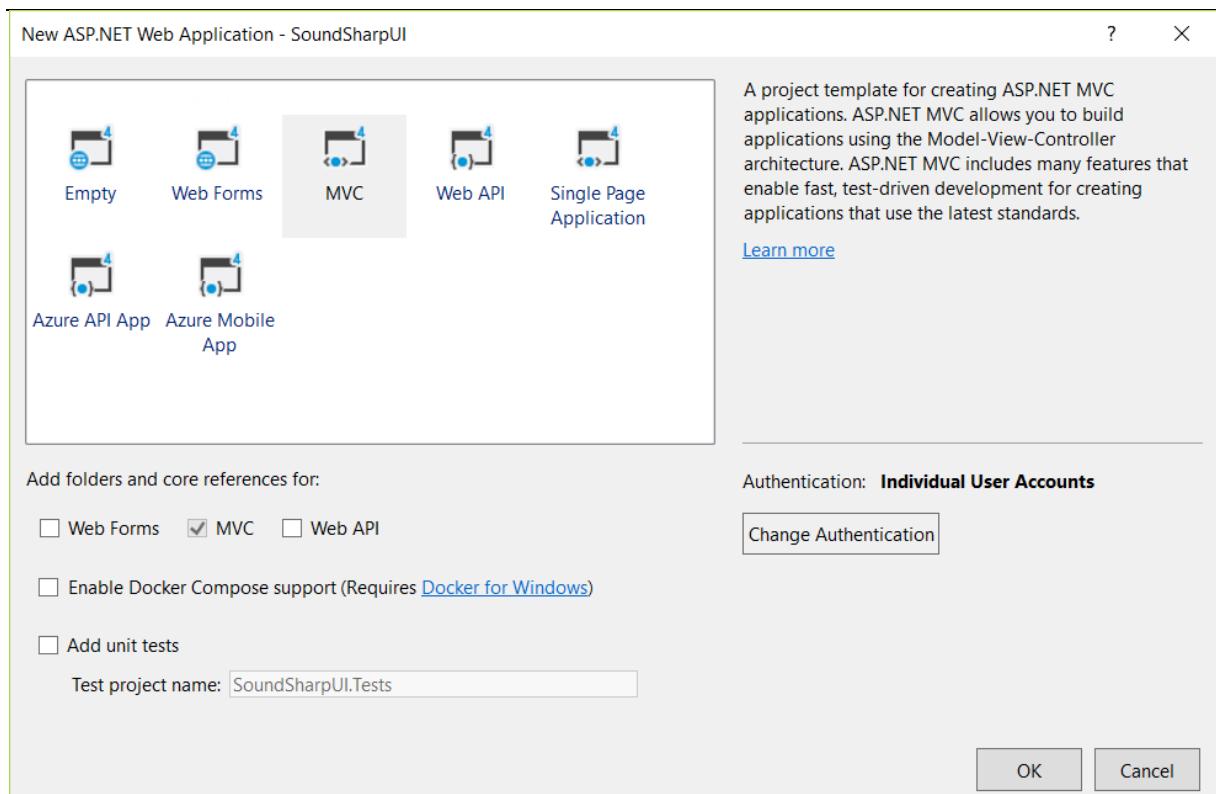
Solution name: SoundsharpMVCWithDB

Framework: .NET Framework 4.5

☒ Create directory for solution

☒ Create new Git repository

OK Cancel

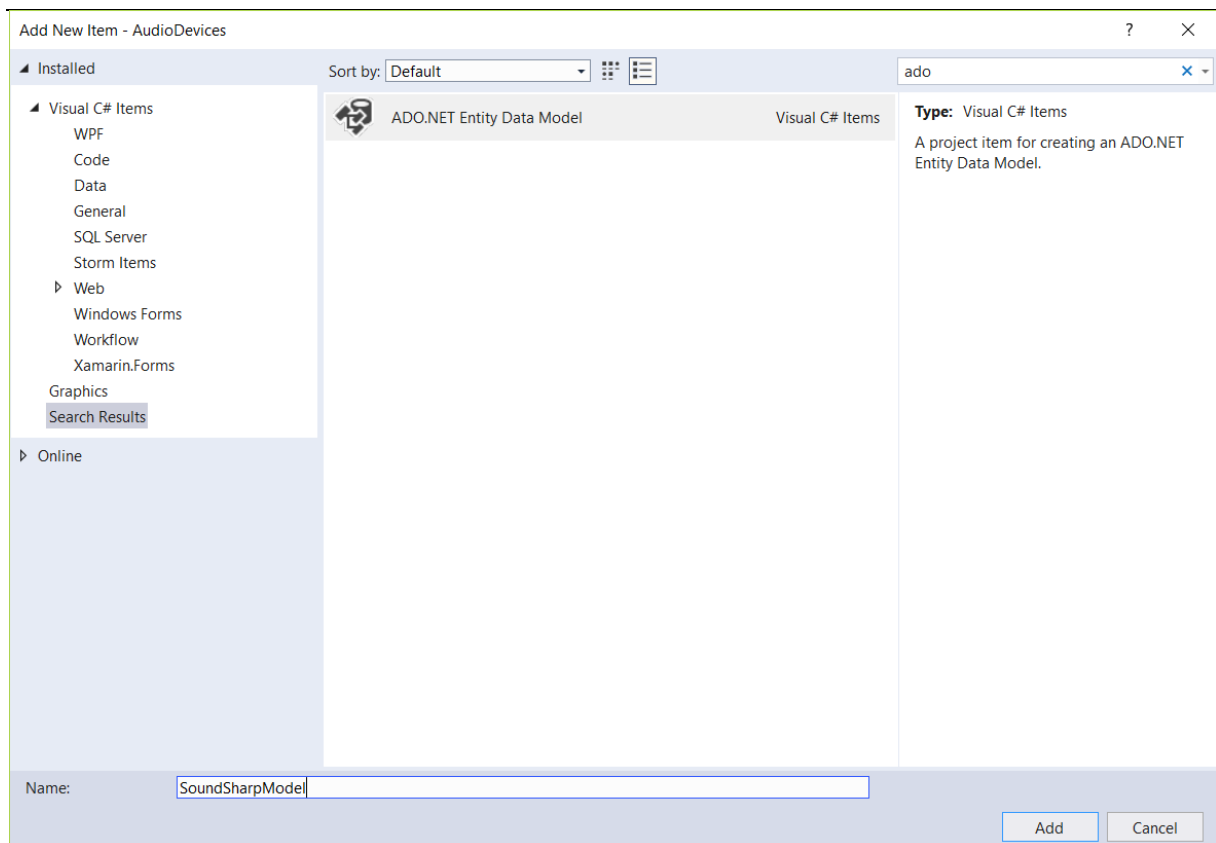


## Stap 2: Class Library aan solution toevoegen

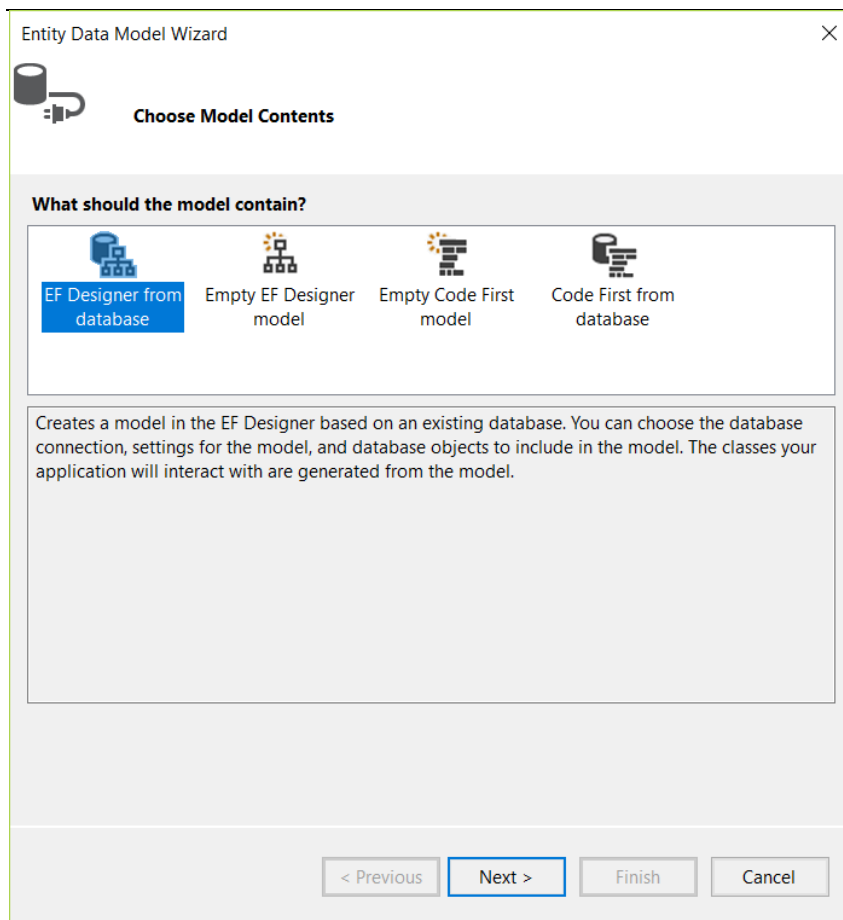
Voeg een tweede project toe aan de solution met als naam **AudioDevices** en kies hiervoor de Class Library template (.NET framework C#). Verwijder het bestand Class1.cs dat standaard wordt aangemaakt.

## Stap 3: Data Model toevoegen aan class library

Voeg aan de class library aan ADO.NET Entity Data Model toe en geef dat als naam **SoundSharpModel**.



Kies voor EF Designer From Database



Maak een nieuwe connectie aan die verwijst naar de eerder gemaakte database:



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
Microsoft SQL Server (SqlClient) Change...

Server name:  
(localdb)\mssqllocaldb Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:  
SoundSharpDB

☐ Attach a database file:  
 Browse...

Logical name:


Advanced...

Test Connection OK Cancel

LET OP: bij server name moet je de locatie invullen van de server waar je eerder de database op hebt gemaakt.



Entity Data Model Wizard



Choose Your Data Connection

Which data connection should your application use to connect to the database?

desktop-e9isk75\localdb#b4a050af.SoundSharpDB.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

metadata=res://\*/SoundSharpModel.csdl|res://\*/SoundSharpModel.ssdl|res://\*/SoundSharpModel.msl;provider=System.Data.SqlClient;provider connection string="data source=(localdb)\mssqllocaldb;initial catalog=SoundSharpDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

SoundSharpDBEntities

< Previous

Next >

Finish

Cancel


Kies **Next** zodat je in het scherm komt waar je de tabellen kunt selecteren. Kies daarin alle tabellen behalve **Sysdiagrams**.

16 | Een ASP.NET MVC website maken





Entity Data Model Wizard

 **Choose Your Database Objects and Settings**

**Which database objects do you want to include in your model?**

- ☒ **Tables**
  - ☒ **dbo**
    - ☒ AudioDevice
    - ☒ Category
    - ☒ CdDiskMan
    - ☒ MemoRecorder
    - ☒ Mp3Player
    - ☐ sysdiagrams
    - ☒ Track
    - ☒ TrackList
    - ☒ TrackListTrack
  - ☐ Views
  - ☐ Stored Procedures and Functions

☐ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

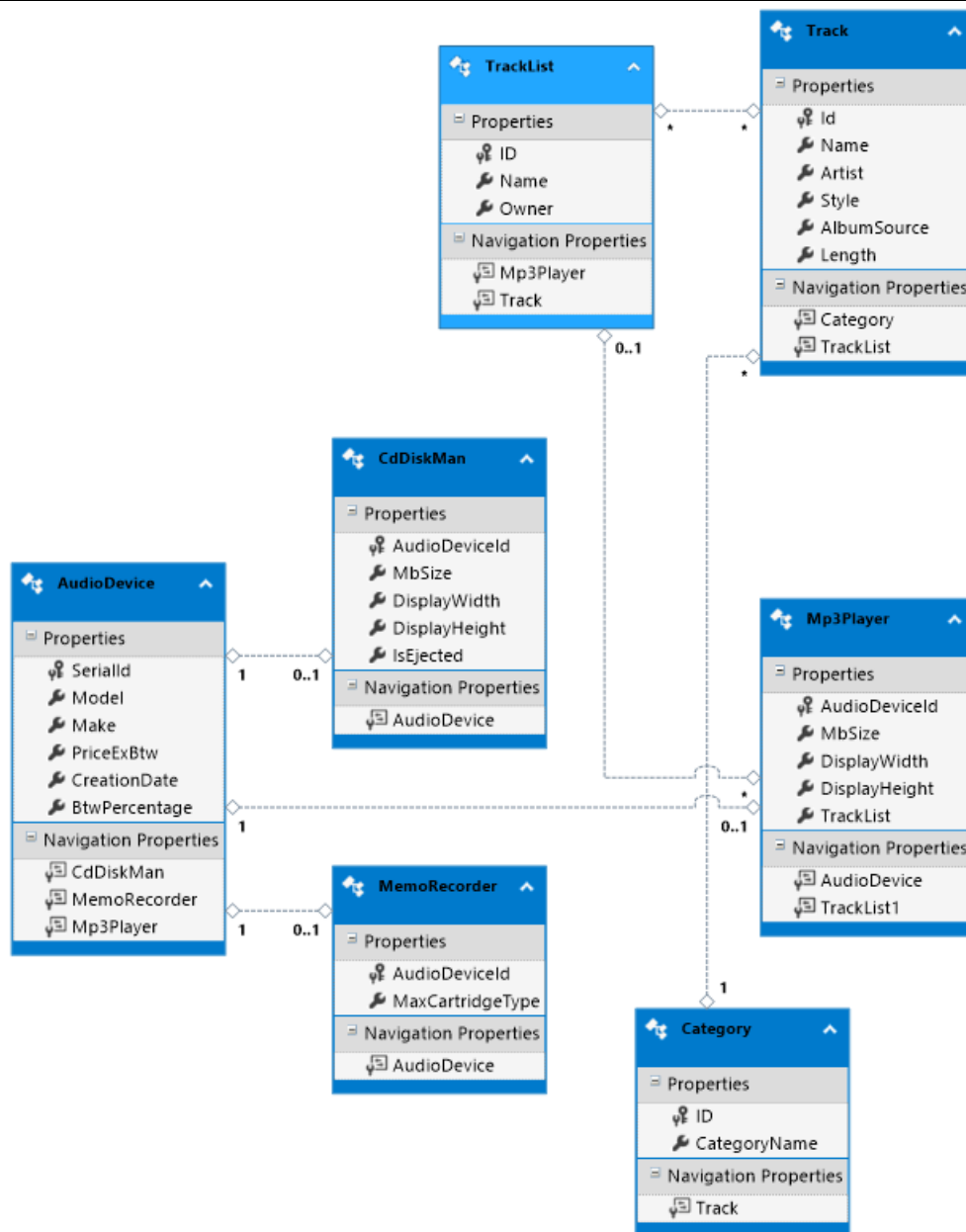
☒ Import selected stored procedures and functions into the entity model

Model Namespace:

SoundSharpDBModel

< Previous   Next >   **Finish**   Cancel

Nadat je op **Finish** hebt geklikt verschijnt het datamodel:



**Nadat het model is gegenereerd dien je eerst de solution opnieuw te bouwen!**

#### Stap 4: Enkele huishoudelijke taken in de solution

Voordat we nu daadwerkelijk kunnen beginnen aan het bouwen van de website dienen we eerst enkele voorbereidingen te treffen.



---

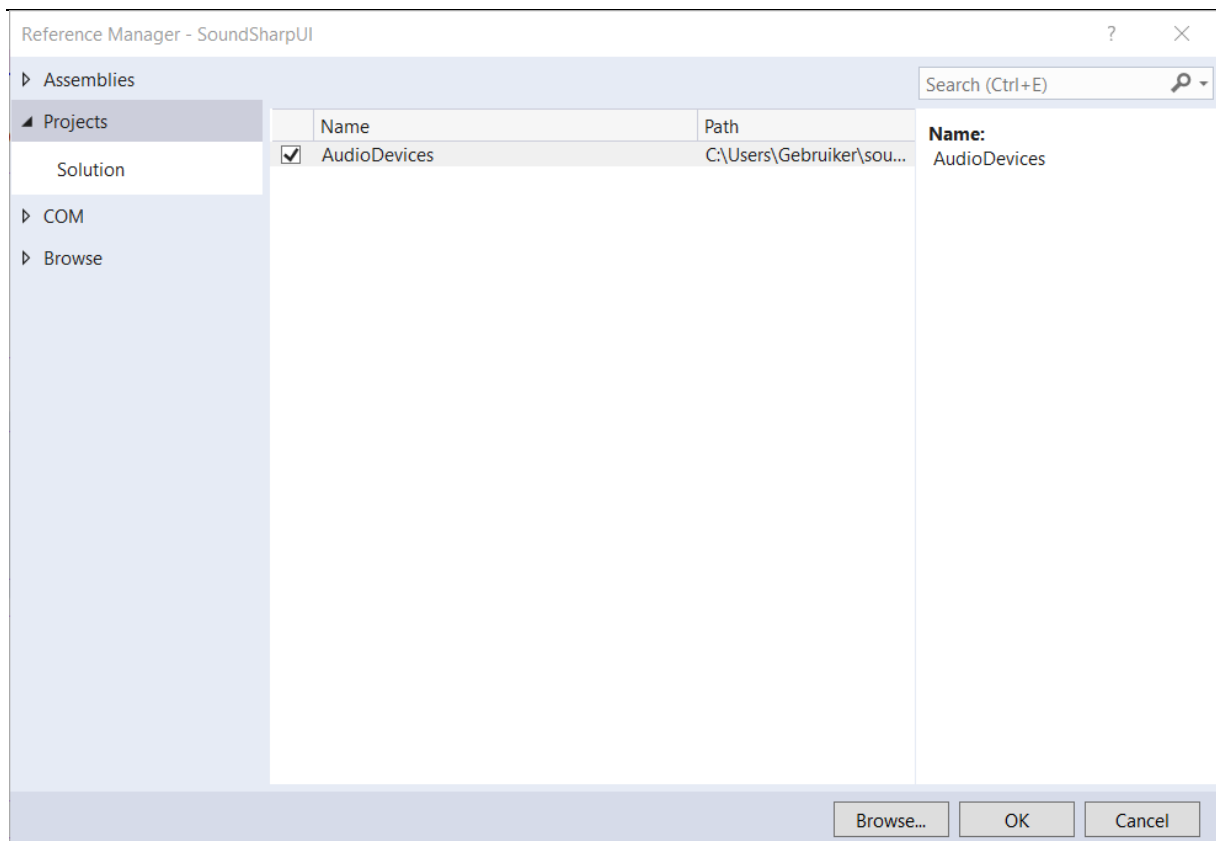
In de class library staat een connectionstring in het bestand *App.config*

```
<connectionStrings>
  <add name="SoundSharpDBEntities"
connectionString="metadata=res://*/SoundSharpModel.csdl|res://*/SoundSharpModel.ssdl|r
es://*/SoundSharpModel.msl;provider=System.Data.SqlClient;provider connection
string=&quot;data source=(localdb)\mssqllocaldb;initial
catalog=SoundSharpDB;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

Deze connectionstring moet worden gekopieerd naar het bestand *Web.config* in je MVC project. In dat bestand bevindt zich al een sectie met connectionstrings. Nadat je de connectionstring hebt gekopieerd ziet het er als volgt uit:

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data
Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-SoundSharpUI-
20190303024703.mdf;Initial Catalog=aspnet-SoundSharpUI-20190303024703;Integrated
Security=True"
  providerName="System.Data.SqlClient" />
  <add name="SoundSharpDBEntities"
connectionString="metadata=res://*/SoundSharpModel.csdl|res://*/SoundSharpModel.ssdl|r
es://*/SoundSharpModel.msl;provider=System.Data.SqlClient;provider connection
string=&quot;data source=(localdb)\mssqllocaldb;initial
catalog=SoundSharpDB;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

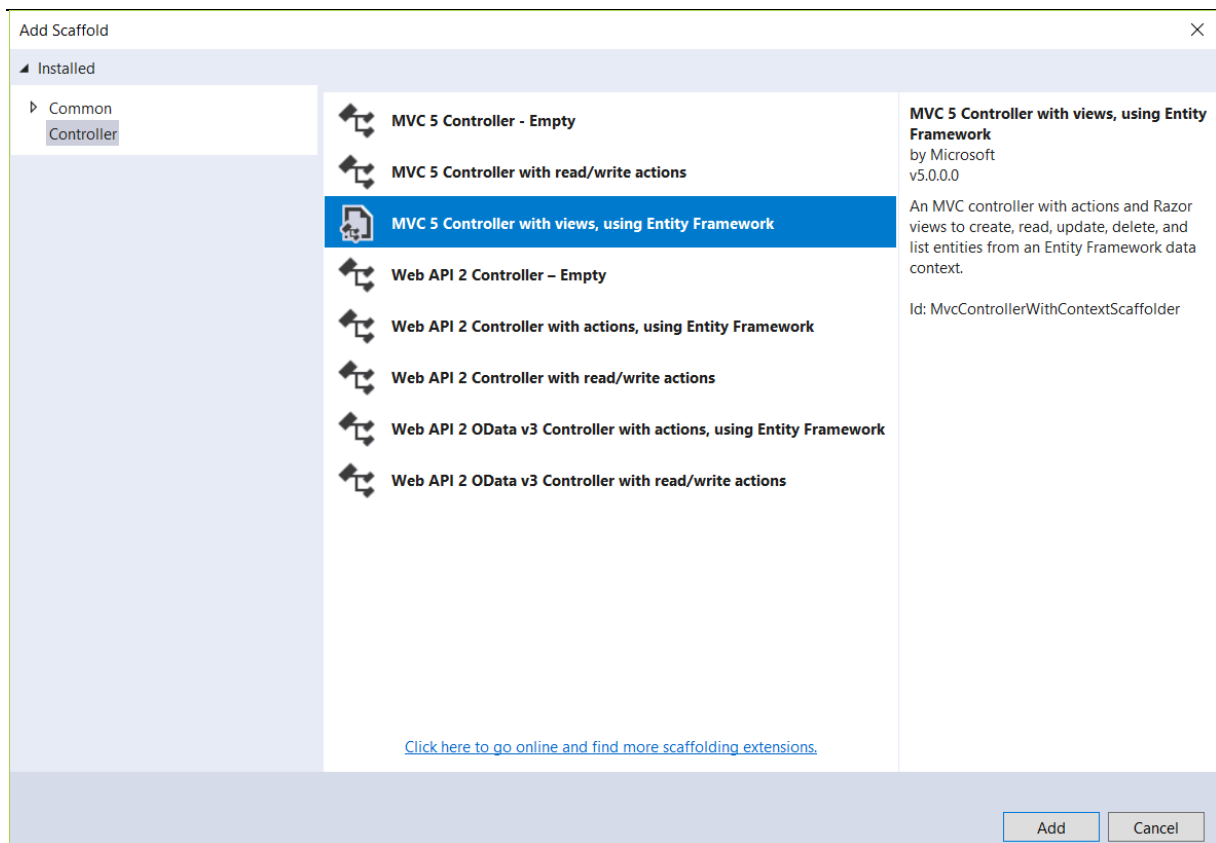
Ook moeten we nog zorgen dat vanuit het MVC project objecten gebruikt kunnen worden uit de class library. Dat doen we door een reference toe te voegen aan het MVC project.



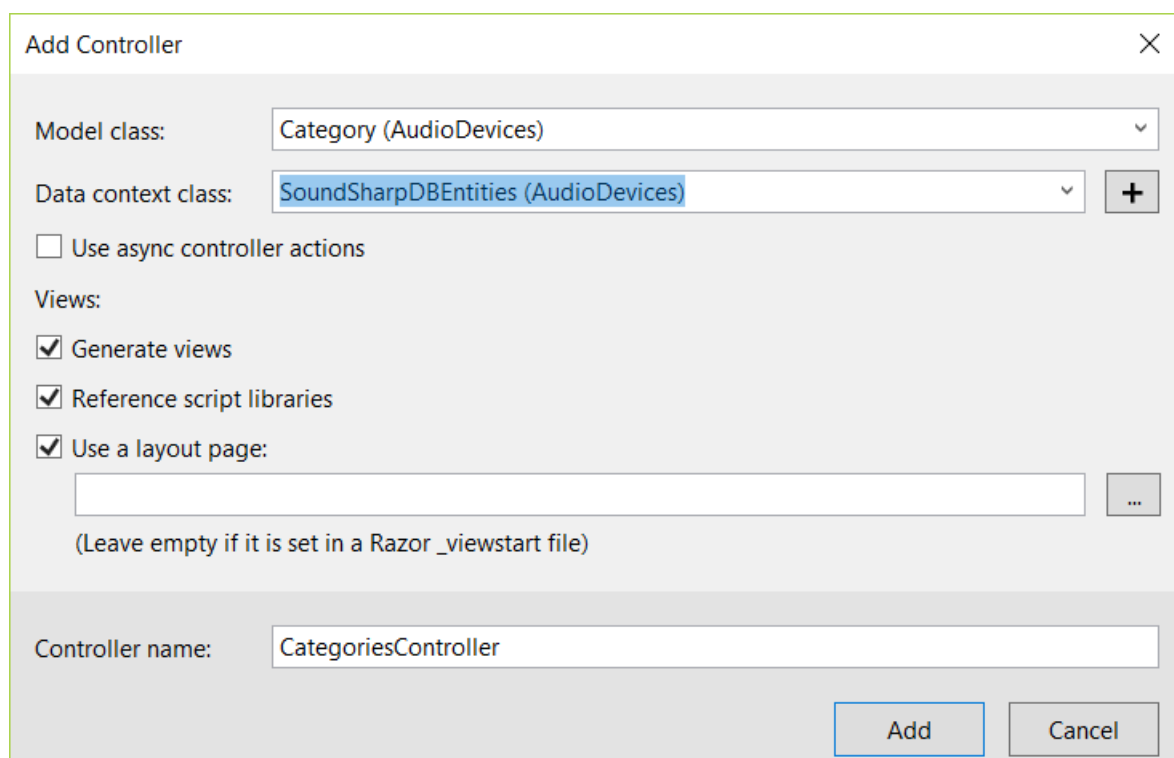
## Stap 5: Categorieën beheren

Voor het beheren van de genres (categorieën) gaan we een controller aanmaken, inclusief alle views om alle CRUD taken uit te voeren: **Create Read Update en Delete**.

Voeg aan de map *Controllers* van je MVC project een nieuwe controller toe met behulp van het Entity Framework.



Kies als Model class voor **Category (AudioDevices)** en stel de Data context class in op *SoundSharpDbEntities* (dit kan bij jou een andere naam zijn). Zorg dat alle vinkjes onder Views aan staan





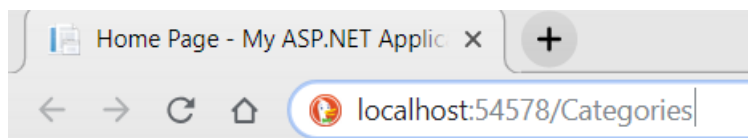
Visual Studio genereert nu alle benodigde views voor het beheren van de categorieën.

Start de applicatie door op [F5] te drukken of klik op de button met de groene knop



Waarschijnlijk wordt nu de homepage gestart. Aangezien we hier nog niets mee hebben gedaan zien we alleen een startpagina met wat informatie over ASP.NET en MVC.

Door echter in de adresbalk de naam van de controller toe te voegen kunnen we de zojuist gemaakt views gebruiken.



Application name		Home	About	Contact
<h1>Index</h1> <a href="#">Create New</a>				
CategoryName				
Ambient	Edit   Details   Delete			
Blues	Edit   Details   Delete			
Country	Edit   Details   Delete			
Disco	Edit   Details   Delete			
Electro	Edit   Details   Delete			
Hardcore	Edit   Details   Delete			
HardRoc	Edit   Details   Delete			
HeavyMetal	Edit   Details   Delete			
Hiphop	Edit   Details   Delete			
Jazz	Edit   Details   Delete			
Immense	Edit   Details   Delete			

Aangezien we zelf geen specifieke action hebben opgegeven zal de action *Index* van de controller *Categories* worden aangeroepen en deze opent de bijbehorende view: een lijst met alle categorieën.



Alle opties om de genres te bewerken zijn van hieruit te gebruiken. Probeer zelf een categorie toe te voegen, een categorie te wijzigen en een categorie te verwijderen.

Probeer de categorie *Pop* eens te verwijderen. Lukt dat? Zo nee, waarom denk je dat er een fout optreedt?

## Stap 6: Tracks beheren

Het kan nu niet zo moeilijk zijn een controller toe te voegen waarmee we tracks kunnen beheren.

Voeg via dezelfde procedure de gewenste controller toe:

The screenshot shows the 'Add Controller' dialog box with the following configuration:

- Model class:** Track (AudioDevices)
- Data context class:** SoundSharpDBEntities (AudioDevices)
- ☐ Use async controller actions
- Views:**
  - ☒ Generate views
  - ☒ Reference script libraries
  - ☒ Use a layout page: (Leave empty if it is set in a Razor \_viewstart file)
- Controller name:** TracksController

The 'Add' button is highlighted in blue.

Kijk of je alle CRUD operaties kunt uitvoeren. Merk op dat bij het toevoegen van een nieuwe track (en bij het wijzigen) automatisch een dropdownlist beschikbaar komt om een keuze te maken uit de lijst met categorieën.



Application name   Home   About   Contact

## Create

Track

Name	<input type="text"/>
Artist	<input type="text"/>
Style	<input type="text" value="Ambient"/>
AlbumSource	<input type="text"/>
Length	<input type="text"/>
<input type="button" value="Create"/>	

[Back to List](#)

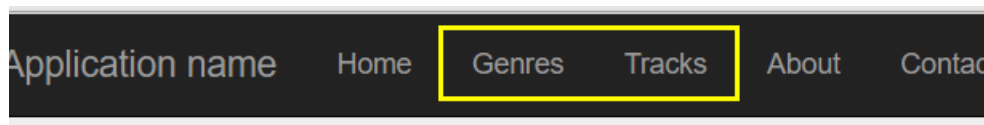
### Stap 7: De menubalk aanpassen

Het is natuurlijk niet handig dat we de acties moeten aanroepen door in de adresbalk de juiste controller aan te roepen. We kunnen het voor de gebruiker een stuk eenvoudiger maken door deze opties toe te voegen aan de menubalk.

Aangezien er al enkel opties in de menubalk zijn aangemaakt kunnen we eenvoudig zelf opties toevoegen. De menubalk is te vinden in de view `_Layout.cshml` welke we kunnen vinden onder de map `Views->Shared`. Voeg de twee links in het gele vierkant toe.

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li>@Html.ActionLink("Home", "Index", "Home")</li>
    <li>@Html.ActionLink("Genres", "Index", "Categories")</li>
    <li>@Html.ActionLink("Tracks", "Index", "Tracks")</li>
    <li>@Html.ActionLink("About", "About", "Home")</li>
    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
  </ul>
```

Start de applicatie opnieuw op en controleer of de twee menuopties zijn toegevoegd.



### Stap 8: Beheertaken alleen voor beheerders

Het is niet moeilijk ervoor te zorgen dat de opties alleen zichtbaar zijn voor ingelogde gebruikers.





```
<ul class="nav navbar-nav">
  <li>@Html.ActionLink("Home", "Index", "Home")</li>
  @if (User.Identity.IsAuthenticated)
  {
    <li>@Html.ActionLink("Genres", "Index", "Categories")</li>
    <li>@Html.ActionLink("Tracks", "Index", "Tracks")</li>
  }
  <li>@Html.ActionLink("About", "About", "Home")</li>
  <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
```

Dit is een eenvoudige oplossing, maar iemand die weet hoe MVC werkt zal alsnog de beheerpagina's kunnen benaderen door de gewenste actie vanuit de adresbalk aan te roepen. Echt veilig is het dus nog niet.

Om ervoor te zorgen dat alleen ingelogde gebruikers beheertaken kunnen uitvoeren hoeven we slechts een kleinigheid toe te voegen aan de eerder gemaakte controllers. Als voorbeeld gaan we het beheren van categorieën en tracks alleen toegankelijk maken voor ingelogde gebruikers.

Open eerst de code van de CategoriesController en voeg direct voor de definitie van de class toe:

[Authorize]

```
namespace SoundSharpUI.Controllers
{
    [Authorize]
    0 references | 0 changes | 0 authors, 0 changes
    public class CategoriesController
    {
        private SoundSharpDBEntities d
```

Indien Visual Studio dit niet herkent dien je *System.Web.Mvc* aan de usings toe te voegen:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using AudioDevices;
```

Als je nu probeert via de adresbalk naar de controller *Categories* te gaan wordt je doorgestuurd naar de loginpagina.



Log in - My ASP.NET Application

localhost:54578/Account/Login?ReturnUrl=%2FCategories

Apps Programming Exerc... Agile Android Develop... Geïmporteerd uit Fi... MS-ITA ROC van

Application name Home About Contact

## Log in.

Use a local account to log in.

Email

Password

☐ Remember me?

[Register as a new user](#)

Aangezien standaard een gebruiker zich zelf kan registreren is het nog niet echt veilig. Het gebruik van rollen om ervoor te zorgen dat alleen bepaalde gebruikers bepaalde taken uit kunnen voeren is niet zo heel moeilijk, maar daar gaan we nu niet verder op in.

Door ook aan de *TracksController* het authorize attribuut toe te voegen kunnen we ook die taken ontoegankelijk maken voor niet-ingelogde gebruikers.

```
namespace SoundSharpUI.Controllers
{
    [Authorize]
    public class TracksController : Controller
    {
        private SoundSharpDBEntities db = new S
```

Door aan een action het attribuut `[AllowAnonymous]` toe te voegen kunnen we ervoor zorgen dat die methode wel beschikbaar is voor niet-ingelogde gebruikers.

```
[AllowAnonymous]
// GET: Categories
public ActionResult Index()
{
    return View(db.Category.ToList());
}

[AllowAnonymous]
// GET: Tracks
public ActionResult Index()
{
    var track = db.Track.Include(t => t.Category);
    return View(track.ToList());
}
```



Als een niet-ingelogde gebruiker in een overzicht een taak kiest waarvoor hij wel ingelogd moet zijn (bijvoorbeeld een nieuw item aanmaken of een item bewerken/verwijderen) dan wordt hij/zij weer naar het inlogvenster gestuurd.

We kunnen er ook voor zorgen dat deze opties niet zichtbaar zijn in de views voor niet-ingelogde gebruikers:

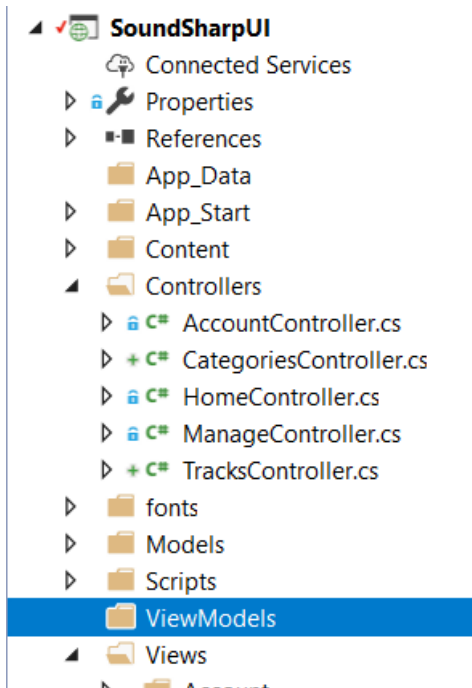
```
@if (User.Identity.IsAuthenticated)
{
    <td>
        @Html.ActionLink("Edit", "Edit", new { id = item.ID }) |
        @Html.ActionLink("Details", "Details", new { id = item.ID }) |
        @Html.ActionLink("Delete", "Delete", new { id = item.ID })
    </td>
}
```

## Stap 9: Devices beheren

Na het voorgaande denk je wellicht dat het eenvoudig is om op vergelijkbare wijze controllers aan te maken voor het beheren van de audiodevices. Dit is echter wat lastiger, omdat het toevoegen van een audiodevice betekent dat je de in 2 tabellen wat moet toevoegen. Om bijvoorbeeld een MemoRecorder toe te voegen moet je een AudioDevice en een MemoRecorder toevoegen. Om dat op te lossen maken we gebruik van een viewmodel.

### Stap 9a: Een ViewModel voor een MemoRecorder

Om het voor onszelf overzichtelijk te houden maken we eerst in het MVC project een nieuwe map aan voor de viewmodels:



In deze map maken we een nieuwe class aan voor een viewmodel voor een MemoRecorder: **vwMemoRecorder**.



```
public class VmMemoRecorder
{
    [Key]
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public int SerialId { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public string Make { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public string Model { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public DateTime CreationDate { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public decimal PriceExBtw { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public double BtwPercentage { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public string MaxMemoCartridgeType { get; set; }
}
```

**LET OP: je moet eerst je project builden voordat je een controller kunt toevoegen.**

Vervolgens maken we op dezelfde manier als hiervoor een controller aan voor de MemoRecorders, waarbij we als Data class het viewmodel gebruiken.

Add Controller

Model class: vwMemoRecorder (SoundSharpUI.ViewModels)

Data context class: SoundSharpDBEntities (AudioDevices)

☐ Use async controller actions

Views:

☒ Generate views

☒ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor \_viewstart file)

Controller name: MemoRecordersController

Add Cancel

Zoals je zult merken verschijnen er diverse foutmeldingen in de controller die wordt aangemaakt. Visual Studio gaat ervan uit dat we een tabel *vwMemoRecorder* hebben in onze database, maar dat is niet juist. Daarom gaan we zelf een lijst met memorecorders maken die we vervolgens naar de view gaan sturen.



Pas de action *Index* als volgt aan:

```
// GET: MemoRecorders
0 references | 0 changes | 0 authors, 0 changes | 0 requests | 0 exceptions
public ActionResult Index()
{
    var memorecorders = new List<VmMemoRecorder>();
    foreach (var mr in db.MemoRecorder)
    {
        // Create a new viewmodel
        var recorder = new VmMemoRecorder()
        {
            Make = mr.AudioDevice.Make,
            Model = mr.AudioDevice.Model,
            CreationDate = mr.AudioDevice.CreationDate,
            PriceExBtw = mr.AudioDevice.PriceExBtw,
            BtwPercentage = (double)mr.AudioDevice.BtwPercentage,
            SerialId = mr.AudioDevice.SerialId,
            MaxMemoCartridgeType = mr.MaxCartridgeType
        };
        // Add the viewmodel to the list
        memorecorders.Add(recorder);
    }
    return View(memorecorders);
}
```

Op vergelijkbare wijze passen we ook *Details* aan:

```
// GET: MemoRecorders/Details/5
0 references | mroesink@rocvantwente.nl, 18 minutes ago | 1 author, 1 change | 0 requests | 0 exceptions
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var recorder = db.MemoRecorder.Find(id);
    if (recorder == null)
    {
        return HttpNotFound();
    }
    var mr = new VmMemoRecorder()
    {
        Make = recorder.AudioDevice.Make,
        Model = recorder.AudioDevice.Model,
        CreationDate = recorder.AudioDevice.CreationDate,
        PriceExBtw = recorder.AudioDevice.PriceExBtw,
        BtwPercentage = (double)recorder.AudioDevice.BtwPercentage,
        SerialId = recorder.AudioDevice.SerialId,
        MaxMemoCartridgeType = recorder.MaxCartridgeType
    };
    return View(mr);
}
```

En ook de Create post action:



```
// POST: MemoRecorders/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 references | mroesink@rocvantwente.nl, 41 minutes ago | 1 author, 1 change | 2 requests | 0 exceptions
public ActionResult Create([Bind(Include = "SerialId,Make,Model,CreationDate,PriceExBtw,BtwPercentage,MaxMemoCartridgeType")]
    VmMemoRecorder vmMemoRecorder)
{
    if (ModelState.IsValid)
    {
        var device = new AudioDevice()
        {
            Make = vmMemoRecorder.Make,
            Model = vmMemoRecorder.Model,
            PriceExBtw = vmMemoRecorder.PriceExBtw,
            BtwPercentage = (decimal)vmMemoRecorder.BtwPercentage,
            CreationDate = vmMemoRecorder.CreationDate,
            SerialId = vmMemoRecorder.SerialId
        };

        var recorder = new MemoRecorder()
        {
            MaxCartridgeType = vmMemoRecorder.MaxMemoCartridgeType,
            AudioDevice = device
        };
        db.AudioDevice.Add(device);
        db.MemoRecorder.Add(recorder);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(vmMemoRecorder);
}
```

En beide Edit actions:

```
// GET: MemoRecorders/Edit/5
0 references | mroesink@rocvantwente.nl, 54 minutes ago | 1 author, 1 change | 2 requests | 0 exceptions
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var recorder = db.MemoRecorder.Find(id);
    if (recorder == null)
    {
        return HttpNotFound();
    }
    var mr = new VmMemoRecorder()
    {
        Make = recorder.AudioDevice.Make,
        Model = recorder.AudioDevice.Model,
        CreationDate = recorder.AudioDevice.CreationDate,
        PriceExBtw = recorder.AudioDevice.PriceExBtw,
        BtwPercentage = (double)recorder.AudioDevice.BtwPercentage,
        SerialId = recorder.AudioDevice.SerialId,
        MaxMemoCartridgeType = recorder.MaxCartridgeType
    };

    return View(mr);
}
```



## De Edit PUT action

```
// POST: MemoRecorders/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 references | mroesink@rocvantwente.nl, 52 minutes ago | 1 author, 1 change | 1 request | 0 exceptions
public ActionResult Edit([Bind(Include = "SerialId,Make,Model,CreationDate,PriceExBtw,BtwPercentage,MaxMemoCartridgeType")]
    VmMemoRecorder vmMemoRecorder)
{
    if (ModelState.IsValid)
    {
        var device = db.AudioDevice.Find(vmMemoRecorder.SerialId);
        device.Make = vmMemoRecorder.Make;
        device.Model = vmMemoRecorder.Model;
        device.PriceExBtw = vmMemoRecorder.PriceExBtw;
        device.BtwPercentage = (decimal)vmMemoRecorder.BtwPercentage;
        device.CreationDate = vmMemoRecorder.CreationDate;
        device.SerialId = vmMemoRecorder.SerialId;

        var recorder = db.MemoRecorder.Find(vmMemoRecorder.SerialId);
        recorder.MaxCartridgeType = vmMemoRecorder.MaxMemoCartridgeType;
        recorder.AudioDevice = device;

        db.Entry(device).State = EntityState.Modified;
        db.Entry(recorder).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(vmMemoRecorder);
}
```

## En tenslotte beide delete actions:

```
// GET: MemoRecorders/Delete/5
0 references | mroesink@rocvantwente.nl, 59 minutes ago | 1 author, 1 change | 1 request | 0 exceptions
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var recorder = db.MemoRecorder.Find(id);
    if (recorder == null)
    {
        return HttpNotFound();
    }
    var mr = new VmMemoRecorder()
    {
        Make = recorder.AudioDevice.Make,
        Model = recorder.AudioDevice.Model,
        CreationDate = recorder.AudioDevice.CreationDate,
        PriceExBtw = recorder.AudioDevice.PriceExBtw,
        BtwPercentage = (double)recorder.AudioDevice.BtwPercentage,
        SerialId = recorder.AudioDevice.SerialId,
        MaxMemoCartridgeType = recorder.MaxCartridgeType
    };

    return View(mr);
}
```





```
// POST: MemoRecorders/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    var recorder = db.MemoRecorder.Find(id);
    var device = db.AudioDevice.Find(id);

    db.MemoRecorder.Remove(recorder);
    db.AudioDevice.Remove(device);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

Als je nu probeert te bouwen zal er een fout verschijnen, omdat Visual Studio probeert een dataset aan te maken voor het viewmodel:

```
0 references | 0 changes | 0 authors, 0 changes
public System.Data.Entity.DbSet<SoundSharpUI.View
```

Verwijder deze regel en probeer opnieuw of je kunt bouwen.



Je hebt nu gezien hoe je de CRUD operaties voor een **MemoRecorder** kunt implementeren.

Aan jou nu de taak deze operaties toe te voegen voor **CdDiscMans** en voor **Mp3Players**.

1. Voeg een viewmodel toe.
2. Voeg een controller toe inclusief de benodigde views.
3. Pas waar nodig de code van de controllers aan, zoals dat ook is gebeurd bij de MemoRecorders.
4. Zorg dat de devices alleen beheerd kunnen worden door een ingelogde gebruiker.

In de bijlage zijn schermafbeeldingen te zien van de diverse schermen.

Mocht je tijdens het testen problemen hebben bij het toevoegen van een device als je probeert een prijs in te voeren met decimalen dan kun je dat oplossen door aan het eind van het bestand `_Layout.cshtml` de volgende code toe te voegen (net voor het eind van de body):





```
<script>
    $(document).ready(function () {
        $('[data-toggle="tooltip"]').tooltip();
    });

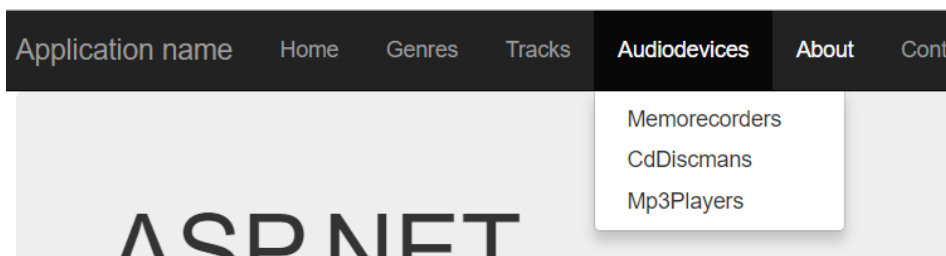
    $.validator.methods.number = function (value, element) {
        return this.optional(element) || /^-
        ?(?:\d+|\d{1,3}(?:\.\d{3})+)?(?:,\d+)?$/ .test(value);
    }
}

</script>
```

### Stap 9d: Een submenu toevoegen aan de menubalk voor audiodevices

Als we alle audiodevices als menuoptie toevoegen aan de menubalk wordt het aantal opties erg groot. Daarom gaan we een submenu toevoegen aan de menubalk, waarin we alle de verschillende audiodevices opnemen:

```
<ul class="nav navbar-nav">
    <li>@Html.ActionLink("Home", "Index", "Home")</li>
    @if (User.Identity.IsAuthenticated)
    {
        <li>@Html.ActionLink("Tracks", "Index", "Tracks")</li>
        <li>@Html.ActionLink("Genres", "Index", "Categories")</li>
        <li class="dropdown">
            <a href="" data-toggle="dropdown" class="dropdown-toggle">Devices</a>
            <ul class="dropdown-menu">
                <li>@Html.ActionLink("Memorecorders", "Index", "MemoRecorders")</li>
                <li>@Html.ActionLink("CdDiscmans", "Index", "CdDiscMans")</li>
                <li>@Html.ActionLink("Mp3 Players", "Index", "Mp3Players")</li>
            </ul>
        </li>
    }
    <li>@Html.ActionLink("About", "About", "Home")</li>
    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
```



**LET OP:** je moet ingelogd zijn om de opties te kunnen zien.



Zorg dat via de menubalk de overzichten met CdDiscMans en Mp3Players ook aangeroepen kunnen worden.

## Stap 10: TracksLists beheren

Tot nu toe hebben we nog geen mogelijkheid om tracks toe te voegen aan een tracklist. Daarvoor moeten we eerst nog een controller aanmaken om de **TrackLists** te kunnen beheren. Met wat we hiervoor hebben uitgevoerd mag dat geen probleem zijn.



Voeg de CRUD operaties voor het beheren van **TrackLists** toe, inclusief de views.

Als je de vorige opdracht goed hebt uitgevoerd ziet het overzicht met TrackLists er als volgt uit:

SoundSharp   Home   About   Contact		
Index		
<a href="#">Create New</a>		
Name	Owner	
My Favorites	Marcel Roesink	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Best of Coldplay	Marcel Roesink	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
© 2019 SoundSharp - Sounds good		

We gaan dit overzicht aanpassen, zodat na iedere tracklist een overzicht komt van de tracks die in die tracklist voorkomen. Daarvoor voegen we in de view een rij aan de tabel toe met één cel. Daarin plaatsen we, met behulp van wat Bootstrap classes, enkele panels waarop de tracks worden weergegeven.

Voeg de volgende code toe aan je view:



```
@foreach (var item in Model) {  
    <tr>  
        <td>  
            @Html.DisplayFor(modelItem => item.Name)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Owner)  
        </td>  
        <td>  
            @Html.ActionLink("Edit", "Edit", new { id=item.ID }) |  
            @Html.ActionLink("Details", "Details", new { id=item.ID }) |  
            @Html.ActionLink("Delete", "Delete", new { id=item.ID })  
        </td>  
    </tr>  
    <tr>  
        <td colspan="3">  
            <div class="row">  
                @foreach(var track in item.Track)  
                {  
                    <div class="col-md-2 col-xs-1 panel panel-default">  
                        <h4>@track.Name</h4>  
                        <h6>@track.Artist</h6>  
                        <p>Genre: @track.Category.CategoryName</p>  
                        <p>Album: @track.AlbumSource</p>  
                        <p>Tijd: @track.Length.ToString()</p>  
                    </div>  
                }  
            </div>  
        </td>  
    </tr>  
}
```

Om het wat te verfraaien voegen we wat CSS toe. Het CSS-bestand is te vinden in de map **Content** en heet standaard **Site.css**.

```
.panel {  
    background-color: aliceblue;  
    margin: 2px;  
    font-size: 0.75em;  
    font-style: italic;  
}
```

Als alles goed verlopen is ziet het overzicht er zo uit:



## Index

[Create New](#)

Name	Owner			
My Favorites	Marcel Roesink		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>	
<div><p><i>Yellow</i></p><p>Coldplay</p><p>Genre: Pop</p><p>Album: Parachutes</p><p>Tijd: 00:04:29</p></div>	<div><p><i>Shiver</i></p><p>Coldplay</p><p>Genre: Pop</p><p>Album: Parachutes</p><p>Tijd: 00:04:59</p></div>	<div><p><i>Maneater</i></p><p>Nelly Futado</p><p>Genre: Pop</p><p>Album: Loose</p><p>Tijd: 00:04:41</p></div>	<div><p><i>Guitar</i></p><p>Prince</p><p>Genre: Pop</p><p>Album:</p><p>Tijd: 00:04:12</p></div>	<div><p><i>Brown Sugar</i></p><p>Rolling Stones</p><p>Genre: Pop</p><p>Album: Sticky Fingers</p><p>Tijd: 00:03:50</p></div>
Best of Coldplay	Marcel Roesink		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>	
<div><p><i>Yellow</i></p><p>Coldplay</p><p>Genre: Pop</p><p>Album: Parachutes</p><p>Tijd: 00:04:29</p></div>	<div><p><i>Shiver</i></p><p>Coldplay</p><p>Genre: Pop</p><p>Album: Parachutes</p><p>Tijd: 00:04:59</p></div>			

© 2019 SoundSharp - Sounds good



De TrackLists kunnen nog niet via het menu benaderd worden. Maak van de menuoptie Tracks een submenu, met daaronder de volgende optie:

- Tracks
- TrackLists

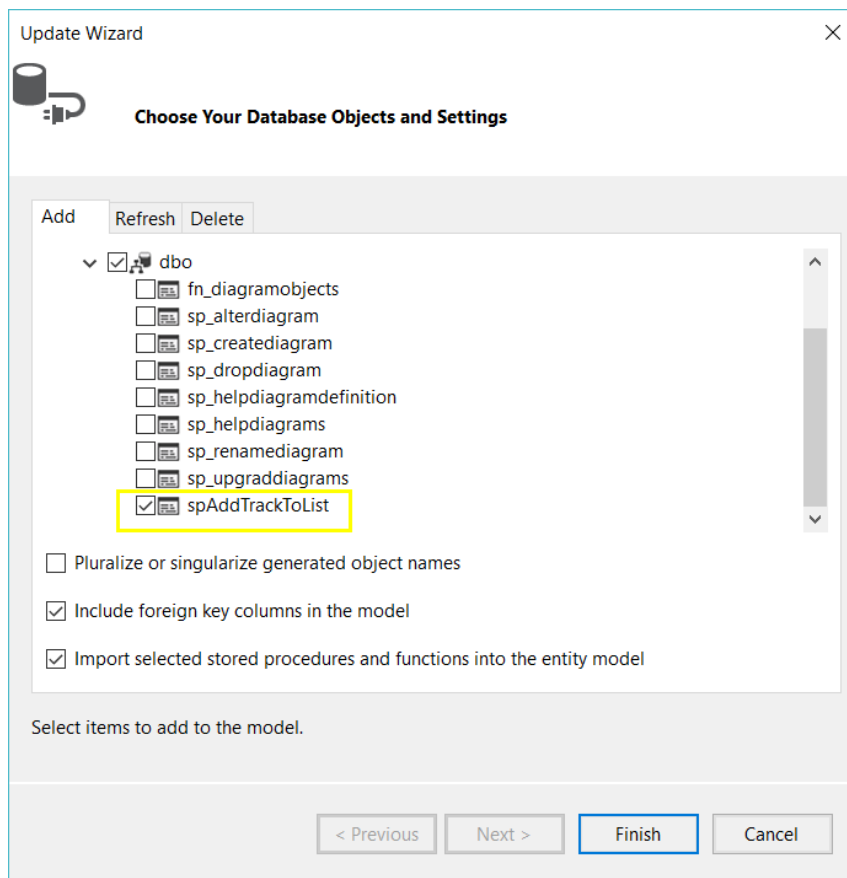
### Stap 11: Tracks aan een tracklist toevoegen

Voor het toevoegen van een track aan een tracklist gaan we gebruik maken van een stored procedure in de database. Gebruik het volgende SQL script om in SQL Server Management Studio een stored procedure aan te maken in de soundsharp database:



```
CREATE PROCEDURE spAddTrackToList
    @TRACKID INT,
    @TRACKLISTID INT
AS
BEGIN
    INSERT INTO TrackListTrack VALUES (@TRACKLISTID , @TRACKID)
END
GO
```

Als dit is gelukt voegen we deze stored procedure toe aan ons data model in het project AudioDevices. Klik ergens met de rechtermuistoets in het diagram en kies *Update model from database*.



**Na iedere wijziging van het model moeten we het project opnieuw bouwen, dus ook nu.**



## Bijlage 1 – Schermafbbeeldingen van CRUD operaties Mp3Players

### Overzicht

#### Index

[Create New](#)

Make	Model	CreationDate	PriceExBtw	BtwPercentage	MbSize	DisplayWidth	DisplayHeight	
Apple	iPod Touch 6	4-3-2018 00:00:00	219,00	21	32000	640	480	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Sandisk	Clip Sports 8GB	9-9-2017 00:00:00	59,99	21	8000	640	480	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Cowon	Plenue D Goud	4-2-2019 00:00:00	284,00	21	32000	1080	640	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Sony	NW-ZX300	10-1-2019 00:00:00	683,00	21	64000	1080	640	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

© 2019 SoundSharp - Sounds good

### Toevoegen

#### Create

VmMp3Player

Make	<input type="text"/>
Model	<input type="text"/>
CreationDate	<input type="text"/>
PriceExBtw	<input type="text"/>
BtwPercentage	<input type="text"/>
MbSize	<input type="text"/>
DisplayWidth	<input type="text"/>
DisplayHeight	<input type="text"/>
TrackList	<input type="text" value="My Favorites"/>
	<input type="button" value="Create"/>

[Back to List](#)



---

## Wijzigen

### Edit

VmMp3Player

<b>Make</b>	<input type="text" value="Sandisk"/>
<b>Model</b>	<input type="text" value="Clip Sports 8GB"/>
<b>CreationDate</b>	<input type="text" value="9-9-2017 00:00:00"/>
<b>PriceExBtw</b>	<input type="text" value="59,99"/>
<b>BtwPercentage</b>	<input type="text" value="21"/>
<b>MbSize</b>	<input type="text" value="8000"/>
<b>DisplayWidth</b>	<input type="text" value="640"/>
<b>DisplayHeight</b>	<input type="text" value="480"/>
<b>TrackList</b>	<input type="text" value="My Favorites"/> ▼
	<input type="button" value="Save"/>

[Back to List](#)



---

Verwijderen

## Delete

Are you sure you want to delete this?

VmMp3Player

---

<b>Make</b>	Sandisk
<b>Model</b>	Clip Sports 8GB
<b>CreationDate</b>	9-9-2017 00:00:00
<b>PriceExBtw</b>	59,99
<b>BtwPercentage</b>	21
<b>MbSize</b>	8000
<b>DisplayWidth</b>	640
<b>DisplayHeight</b>	480
<b>TrackList</b>	1

Delete

| [Back to List](#)

---

© 2019 SoundSharp - Sounds good





## Bijlage 2 – Schermafbbeeldingen van CRUD operaties CdDiscmans

### Overzicht

SoundSharp	Home	Tracks	Genres	Devices	About	Contact	Hello mroesink@rocvantwente.nl!	Log off
------------	------	--------	--------	---------	-------	---------	---------------------------------	---------

### Index

[Create New](#)

Make	Model	CreationDate	PriceExBtw	BtwPercentage	MbSize	DisplayWidth	DisplayHeight	IsEjected	
Soundmaster	CD9220	8-5-2018 00:00:00	48,00	21	700	640	480	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Denver	DM-24	12-12-2015 00:00:00	24,99	21	700	640	480	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Grundig	CDP 6600 Pink	10-10-2018 00:00:00	52,00	21	700	1080	640	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Sony	C60HF	9-10-2011 00:00:00	10,99	21	700	1080	640	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

© 2019 SoundSharp - Sounds good

### Toevoegen

## Create

VmCdDiscMan

Make	<input type="text"/>
Model	<input type="text"/>
CreationDate	<input type="text"/>
PriceExBtw	<input type="text"/>
BtwPercentage	<input type="text"/>
MbSize	<input type="text"/>
DisplayWidth	<input type="text"/>
DisplayHeight	<input type="text"/>
IsEjected	<input type="checkbox"/>
	<input type="button" value="Create"/>

[Back to List](#)



---

## Wijzigen

# Edit

VmCdDiscMan

---

<b>Make</b>	<input type="text" value="Grundig"/>
<b>Model</b>	<input type="text" value="CDP 6600 Pink"/>
<b>CreationDate</b>	<input type="text" value="10-10-2018 00:00:00"/>
<b>PriceExBtw</b>	<input type="text" value="52,00"/>
<b>BtwPercentage</b>	<input type="text" value="21"/>
<b>MbSize</b>	<input type="text" value="700"/>
<b>DisplayWidth</b>	<input type="text" value="1080"/>
<b>DisplayHeight</b>	<input type="text" value="640"/>
<b>IsEjected</b>	<input type="checkbox"/>
	<input type="button" value="Save"/>

[Back to List](#)



---

Verwijderen

## Delete

Are you sure you want to delete this?

VmCdDiscMan

---

<b>Make</b>	Sony
<b>Model</b>	C60HF
<b>CreationDate</b>	9-10-2011 00:00:00
<b>PriceExBtw</b>	10,99
<b>BtwPercentage</b>	21
<b>MbSize</b>	700
<b>DisplayWidth</b>	1080
<b>DisplayHeight</b>	640
<b>IsEjected</b>	<input type="checkbox"/>

Delete

| [Back to List](#)

---

© 2019 SoundSharp - Sounds good