

AD – Routing & Views

Doelstellingen van dit labo

- Werken met routing-parameters
- Een nieuwe controller kunnen toevoegen aan een MVC-applicatie
- Gegevens doorgeven vanuit een controller naar een view
- ViewModels aanmaken en gebruiken bij het genereren van een view
- Strongly-typed views kunnen maken
- De basis-syntax van de Razor view-engine kunnen toepassen

Muziek App

Set-up

Maak een nieuw MVC-project aan met als projectnaam “Muziek App”. Download het zip-bestand van Toledo en pak het uit op jouw computer. Kopieer en plak vervolgens de inhoud van de codebestanden in de folder *Models* van het project.

Voor deze opgave krijg je de code van de volgende klassen:

- Album:
 - Bevat de gegevens van een muziekalbum (CD)
- Song:
 - Bevat de gegevens van één liedje op een album
- AlbumRepository^(*):
 - Bevat een interne lijst met gegevens van een aantal albums. Daarnaast bevat deze klasse methoden voor het opvragen van albums.

() Doorheen het vak zullen we nog vaker gebruik maken van **repository's**. Een repository kan je beschouwen als een abstractie van een databron (bv.: een databank).*

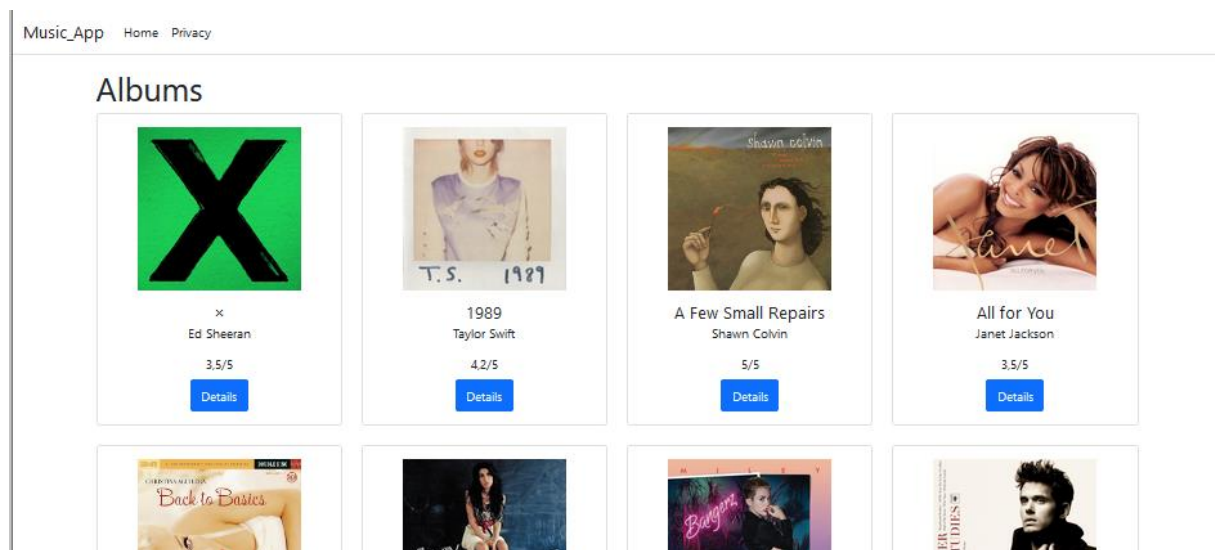
Opgave

Overzicht

Voor deze opgave maak je een eenvoudige webapplicatie waarin je een overzicht toont van een aantal muziekalbums (CD's). De gebruiker kan de details van een specifieke CD opvragen via een details-pagina.

Maak een nieuwe controller aan met de naam **AlbumController**. Voorzie een action met de naam “Index”. Deze action stelt de overzichtspagina met alle albums voor. Maak binnen deze action gebruik van de repository-klasse (zie set-up) om een lijst met alle albums op te vragen en door te geven aan de bijhorende View.

Implementeer voor de Index-pagina een View die er als volgt uitziet:



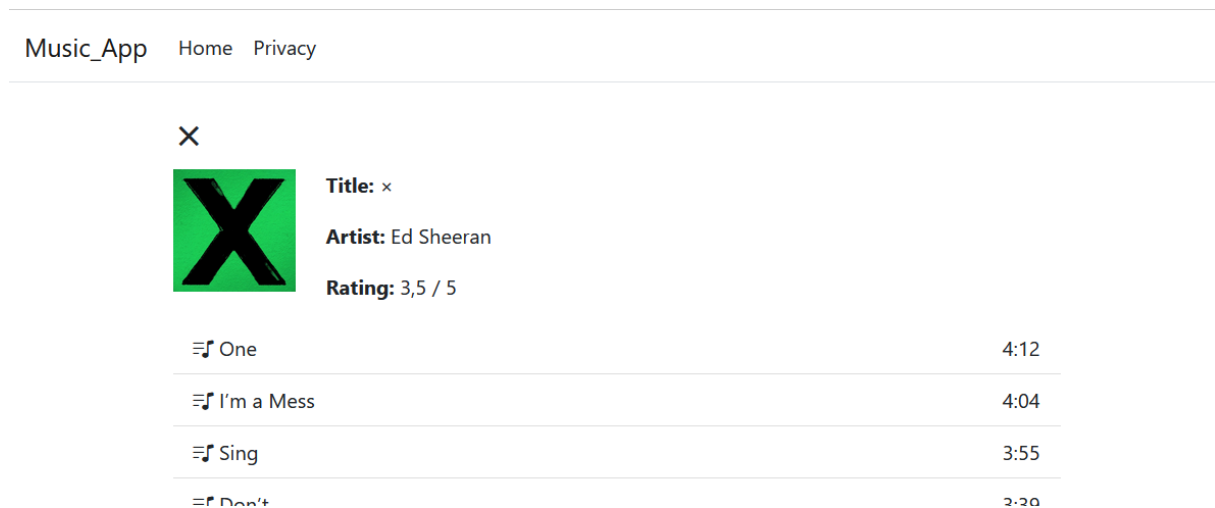
Tips:

- De verschillende albums worden voorgesteld dmv cards. Raadpleeg de [documentatie](#) van Bootstrap om de mogelijkheden van deze component te ontdekken.
- De ImageURL bevat de URL van de cover-afbeelding van het album

Details

Voeg vervolgens een nieuwe action-method toe aan de *AlbumController* voor het opvragen van de Details-pagina. Deze pagina zal de gegevens van één specifiek album weergeven.

Probeer onderstaande View zo goed mogelijk na te bouwen:



- Opmerkingen:
 - Maak voor de lijst met liedjes gebruik van een [list group](#)-component
 - Probeer een icon toe te voegen uit de [Bootstrap Icon-library](#). Voeg een [link](#) toe naar deze library in het `_Layout.cshtml`-bestand om deze icons te kunnen gebruiken.
 - De duur van een liedje kan je opvragen met behulp van de `Length`-property uit de klasse `Song`. **Opgelet:** de lengte wordt uitgedrukt in seconden, je moet deze zélf

converteren naar minuten en seconden. Zorg ervoor dat seconden altijd met twee cijfers weergegeven worden (bv.: 4:04 i.p.v. 4:4)

Zorg er vervolgens voor dat je vanuit de overzichtspagina kunt doorverwijzen naar de details-pagina van een specifiek album.



Opgelet: de Details-knop is een link (`<a>`-element) dat opgemaakt is als een button. Raadpleeg de [documentatie](#) om deze knop na te bouwen.

Om een dynamische link te genereren, kan je in Razor gebruik maken van zogenaamde *tag-helpers*.

Hieronder vind je een voorbeeld van een tag-helper voor het genereren van een link (`<a>`-element):

```
<a asp-controller="Student" asp-action="Details" asp-route-id="3">Details</a>
```

- **asp-controller:** verwijst naar de naam van de **controller** waarvoor de URL in de link moet gegenereerd worden
- **asp-action:** verwijst naar de naam van de **action** binnen deze controller waarvoor de URL in de link moet gegenereerd worden
- **asp-route-id:** waarde van de extra route-value die als parameter moet meegestuurd worden in de URL. De prefix `...-id` verwijst naar de naam van de route-value.

```
<a asp-controller="Student" asp-action="Details" asp-route-id="3">Details</a>

public IActionResult Details(int id) {
    //...
}
```

Indien de routing-value (en dus de naam van de parameter in de Action-method) een andere naam heeft, kan je de prefix `"-id"` in het `asp-route-id` vervangen door de naam van je parameter.

```

<a asp-controller="Student" asp-action="Details"
    asp-route-username="john">Details</a>

public IActionResult Details(string username) {
    //...
}

```

Maak bij het genereren van de links gebruik van tag-helpers. De naam van de controller (**asp-controller**) en de action (**asp-action**) kan je vast instellen op de naam van de Controller en de naam van de action die je gemaakt hebt voor het weergeven van de details. De parameter die het id meegeeft, dien je voor elk album dynamisch te genereren.




Afwerking

- De MVC project-template is een goed vertrekpunt, maar bevat veel overbodige code. Verwijder de Controller en View(s) die je voor deze applicatie niet nodig hebt.
- Voeg tevens een link naar de overzichtspagina toe in de navigatiebalk en verwijder links naar pagina's die niet meer bestaan.

Uitdaging (Extra)

Zorg ervoor dat in het overzicht de rating weergegeven wordt m.b.v. sterretjes in plaats van numeriek.

Albums

 <p>×</p> <p>Ed Sheeran</p> <p>★★★★☆</p> <p>Details</p>	 <p>1989</p> <p>Taylor Swift</p> <p>★★★★★</p> <p>Details</p>	 <p>A Few Small Repairs</p> <p>Shawn Colvin</p> <p>★★★★★</p> <p>Details</p>
--	---	--

Rond af naar halve sterren (bv.: een rating van 3,2 → 3 sterren; 3,7 → 3,5 sterren; 3,8 → 4 sterren).

Tip: ook hier kan je gebruik maken van de Bootstrap Icons-library.

Comme Chez Swa (deel 2)

Context

Voor dit labo, werk je verder aan de website voor het restaurant *Comme Chez Swa*. Geïnteresseerde gasten moeten via de website van Comme Chez Swa de menukaart van het restaurant kunnen raadplegen. Comme Chez Swa biedt drie verschillende menu's aan: een ontbijtmenu, een lunchmenu en een suggestiemenu. Elk menu op een overzichtelijke en gebruiksvriendelijke manier kunnen opgevraagd worden. Om vegetariërs te helpen bij het maken van hun keuze, worden de vegetarische gerechten op een andere manier weergegeven dan de andere gerechten.

Comme Chez Swa Home Menu Over Ons	
<h3>Lunchmenu</h3> <ul style="list-style-type: none">• Stoofvlees met frietjes: € 14,95• Vol-au-vent met puree: € 16,50• <i>Vegetarische Luikse Salade (vegetarisch): € 12,90</i>	<h3>Onze Menu's</h3> <ul style="list-style-type: none">• Ontbijtmenu• Lunchmenu• Suggestiemenu
© 2021 - Comme Chez Swa	

Opgave

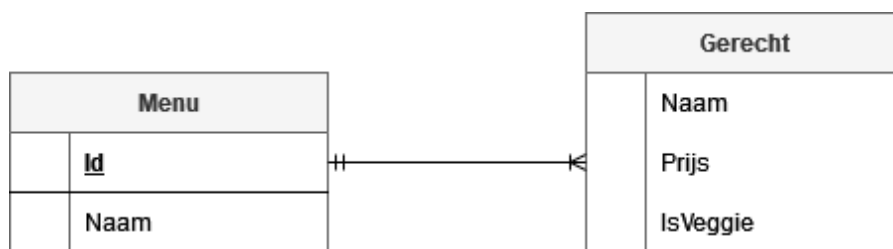
Vorbereiding

Download het zip-bestand met de start-bestanden van Toledo. In het mapje “Models” vind je de code voor drie model-klassen:

- Menu
- Gerecht
- MenuRepository

Kopieer deze klassen en plaats ze in de Models-folder van je project.

Hieronder vind je de schematische weergave (ERD) van de twee Model-klassen *Menu* en *Gerecht*:



Elk Menu heeft een “id”. Dit is een soort van “code” die aangeeft over welk menu het gaat. De verschillende id’s zijn:

- “ONT” → ontbijtmenu
- “LUNCH” → lunchmenu
- “SUGG” → suggestiemenu

Menu weergeven

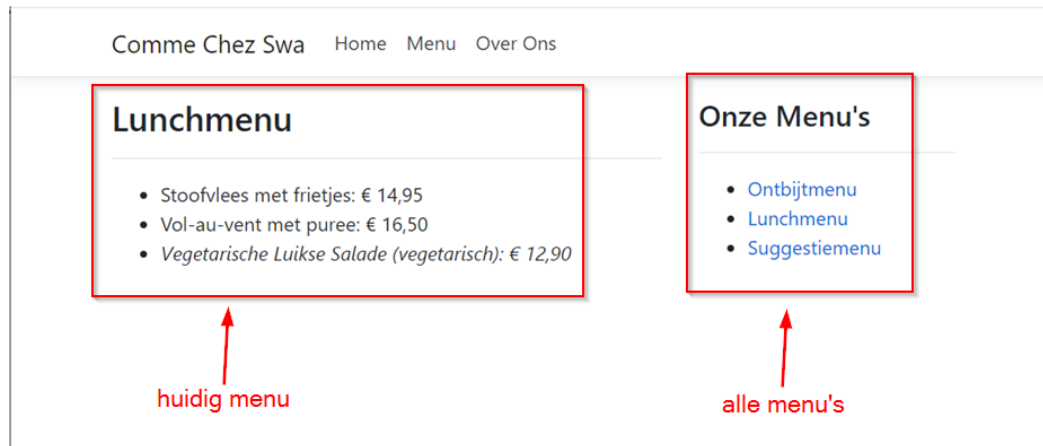
Controller

Voeg een nieuwe Controller toe met de naam **MenuController**, die de requests afhandelt met betrekking tot de Menukaart (rechtermuisknop op Controllers-folder > Add > Controller... en selecteer vervolgens *MVC Controller-Empty*).

Maak binnen deze controller een nieuwe Action-method voor het weergeven van een specifiek menu, **op basis van het gewenste type** (voorzie de Action-method hiervoor van de nodige parameter). Aan de hand van het id van een menu, kan je met behulp van de *MenuRepository*-klasse het menu ophalen van dat type.

ViewModel

In de view wordt zowel het gewenste menu weergegeven als een overzicht van alle beschikbare menu's:



Maak hiervoor een ViewModel-klasse die property's bevat om beide gegevens te groeperen in één object, zodat je dit als één geheel aan de view kunt meegeven.

View

Maak voor het weergeven van het menu een bijhorende view aan. Creëer hiervoor eerst een nieuwe sub-folder in de map "Views", met dezelfde naam als de Controller. Voeg in deze map vervolgens een nieuwe view toe (rechtermuisknop op folder > Add > View... en kies vervolgens *Razor View – Empty*). Zorg ervoor dat de view dezelfde naam heeft als de Action-method die deze view zal terugsturen.

Begin met het instellen van de **titel** van de pagina (**tip**: baseer je hiervoor op de reeds bestaande pagina's in het project).

Maak in de view gebruik van het **ViewModel** dat via de controller wordt doorgegeven aan deze view (m.a.w. dit is een strongly-typed view). Maak gebruik van Razor-syntax om de titel van het huidige menu weer te geven, gevolgd door een lijst met de gerechten van dit menu.

Geef elk vegetarisch gerecht als volgt weer: <naam gerecht> (vegetarisch): <prijs>. Zorg ervoor dat deze gerechten cursief weergegeven worden. Maak hiervoor gebruik van de correcte Bootstrap CSS-class.

Andere gerechten geef je gewoon weer (niet-cursief), en worden uiteraard niet gevolgd door (vegetarisch).

Tip: om de prijs correct weer te geven, kan je gebruik maken van string-formatting (zie <https://docs.microsoft.com/en-us/dotnet/standard/base-types/standard-numeric-format-strings#the-currency-c-format-specifier>)

Aan de rechterkant van het scherm, wordt een lijst met links naar alle menu-types getoond. Maak hiervoor een nieuwe list aan, waarin je voor elk menu een link genereert om dit menu te raadplegen. Haal daarvoor de lijst met alle beschikbare menu's uit het ViewModel.

Afwerking

Voorzie een nieuw item in de navigatiebalk voor de menu-pagina van het **LUNCH**-menu.