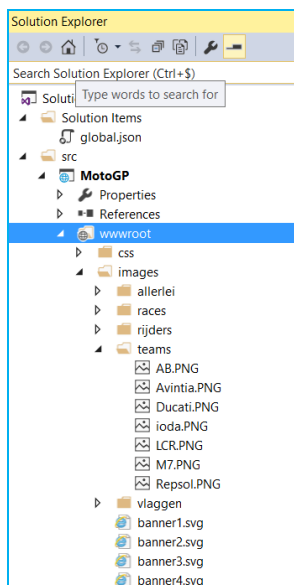


Oefening MotoGP – deel 1

Tijdens deze oefening creëren we een website voor *MotoGP*, het Formule 1 circuit voor motoren. In het eerste deel van de oefening creëren we het project, we passen de default template aan en we maken al heel wat controllers en views aan. Deze views doen spijtig genoeg nog niet heel veel, maar dat komt later wel.

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

Voor het ontwikkelen van de website heb je een aantal bestanden nodig. Kopieer de mappen *allerlei*, *races*, *rijders*, *teams* en *vlaggen* (met hun inhoud) naar de map *images* onder *wwwroot*:



We overschrijven ook *site.css* in de *css*-map met de *site.css* van de *MotoGP* oefening.

1. *_NewLayout.cshtml*

Maak een extra *layout template page* bij met als naam *_NewLayout.cshtml*. De inhoud is

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewData["Title"] - MotoGP</title>
  <link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
  <div>
    @RenderBody()
  </div>
</body>
</html>
```

Merk op dat er een link gelegd wordt naar de *css* van de site. De *~* is een shortcut naar de map *wwwroot*.

2. De homepage

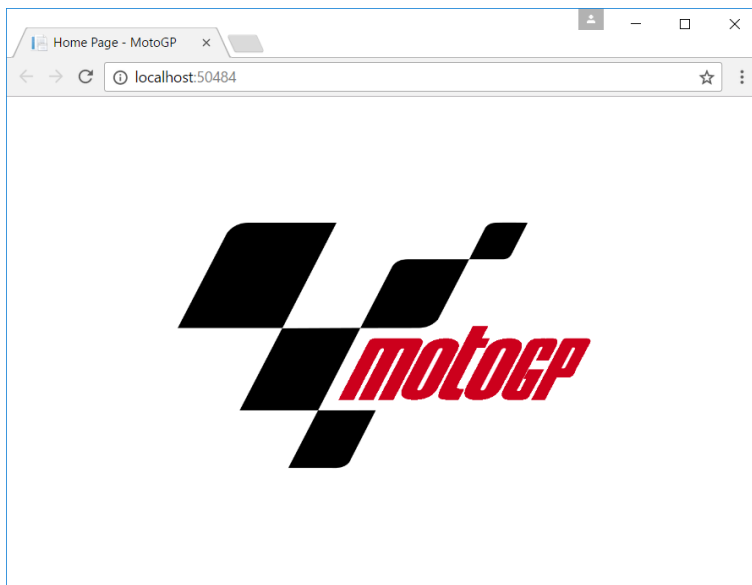
We passen nu eerst de *Index-view* van de *Home-controller* aan. Doe alle code die erin zit weg. Geef nu volgende programmacode in:

```
@{
    Layout = "~/Views/Shared/_NewLayout.cshtml";

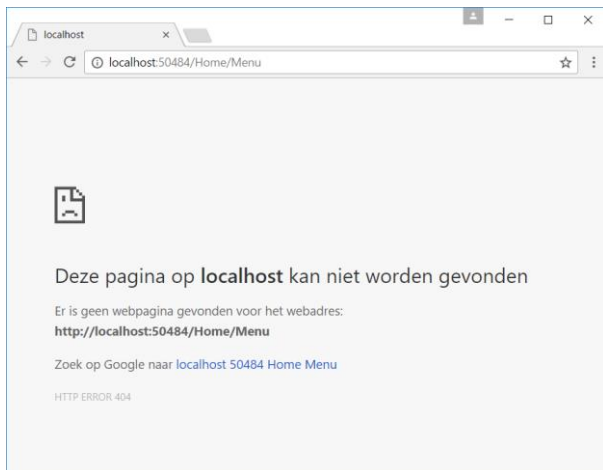
    ViewData["Title"] = "Home Page";
}

<style>
    .centered {
        position: fixed;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
    }
</style>
```

De eerste lijn zorgt ervoor dat niet *_Layout.cshtml* gebruikt wordt als *layout-template*, maar wel de template van daarnet. We stellen ook de *Title* van de pagina in. Met een *inline-style* wordt een klasse *.centered* gedefinieerd waarmee je zo meteen een afbeelding kan centeren. Vervolledig nu zelf de *Index-view* (wanneer je op *Ctrl-F5* drukt, krijg je):



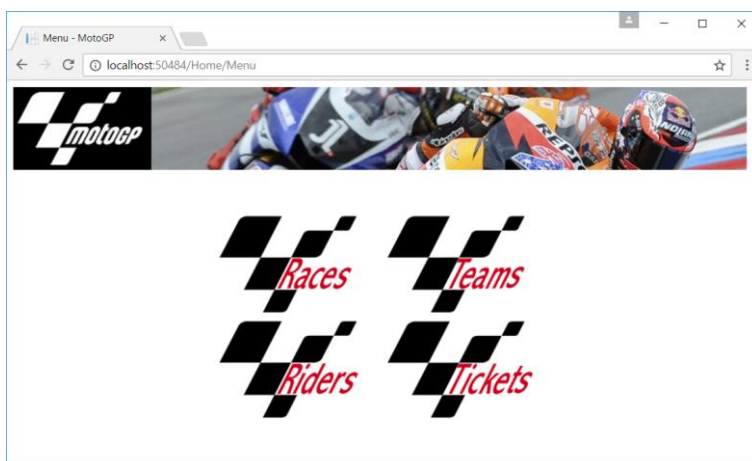
Je ziet een gecentreerde afbeelding met het logo van *MotoGP*. De breedte van de afbeelding is 400. Gebruik een **tag helper** om van die afbeelding een hyperlink te maken. Als je er op klikt, dan spring je naar de *Menu-methode* van de *Home-controller* (die bestaat voorlopig nog niet):



3. Het hoofdmenu

Maak in de *Home-controller* een extra methode *Menu* (*About* en *Contact* mogen weg). Genereer in de methode een willekeurig getal tussen 1 en 4 (beiden inclusief). Stop dit getal in de *DataView-dictionary* met als key *BannerNr*. Roep daarna de *view* op.

Maak nu de *view* aan. Deze *view* gebruikt ook *_NewLayout.cshtml* als layout template:



Bovenaan zie je een banner (een *Image*). Je beschikt hiervoor over de bestanden *banner1.jpg*, *banner2.jpg*, *banner3.jpg* en *banner4.jpg*. Al deze bestanden vind je terug in de *allerlei* map. Welke afbeelding getoond wordt, wordt beslist a.h.v. de waarde van het willekeurig getal uit de *DataView-dictionary*. Wanneer je dus de pagina een aantal keer na mekaar opvraagt, krijg je steeds een andere banner te zien.

De verschillende keuzemogelijkheden van de site worden opgesomd door de vier figuren. In volgende tabel zie je de namen van de figuren en de *Controllers/Methodes* waarnaar je springt als je er op klikt (gebruik een *tag helper*):

figuur	springen naar
~/images/allerlei/races.png	<i>Info/ListRaces</i>
~/images/allerlei /teams.png	<i>Info/ListTeams</i>
~/images/allerlei /riders.png	<i>Info/ListRiders</i>
~/images/allerlei /tickets.png	<i>Shop/OrderTicket</i>

Probeer deze vier hyperlinks mooi te schikken in het midden van de pagina. Gebruik hiervoor je *css* kennis.

In de *title* tab toon je *Menu - MotoGP*.

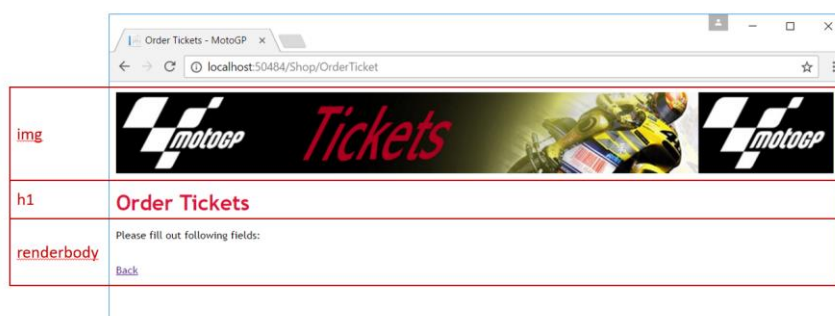
4. *_Layout.cshtml*

We passen nu de *default layoutpage* aan. Doe alle code weg en kopieer de code van *_NewLayout.cshtml* naar *_Layout.cshtml*.

Definieer nu een *Razor* variabele *banners* met daarin alle mogelijke banners die in de applicatie gebruikt kunnen worden:

```
</head>
@{
    var banners = new List<String>
    { "bannerRaces.jpg", "bannerRiders.jpg", "bannerTeams.jpg", "bannerTickets.jpg" };
}
<body>
```

In het *body*-gedeelte plaats je een *img*. Welke afbeelding getoond wordt in die *img*, hangt af van de waarde van *BannerNr* die je haalt uit de *ViewData-dictionary* (*BannerNr* zal dus een waarde zijn tussen 0 en 3). Onder die banner toon je een *h1* met daarin de *Title* van de pagina. Dit is het resultaat als *BannerNr* de waarde 3 heeft (je kan straks pas uittesten als je ook de views maakt):



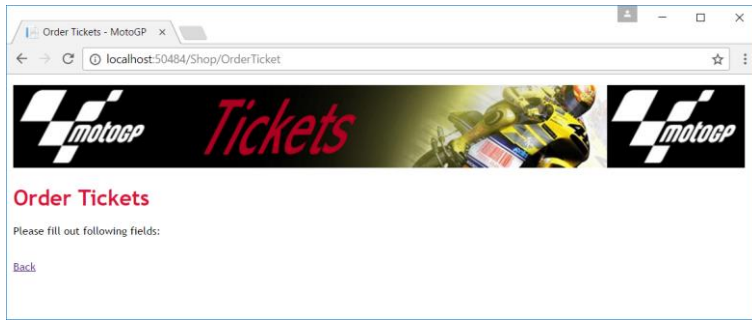
Zorg dat je kan klikken op de banner. Je springt dan naar de homepage.

5. *Tickets bestellen*

Via de applicatie kunnen tickets besteld worden voor de *MotoGP* races. Deze reservaties worden uiteraard in een database weggeschreven, maar dat doe je pas later. Voorlopig maken we alleen al de controller en de views aan.

Maak een nieuwe controller *Shop* (respecteer de naamgeving voor controllers). Maak hierin een methode *OrderTicket* (de *Index*-methode mag je wegdoen). Plaats *BannerNr* op 3 en roep de view op.

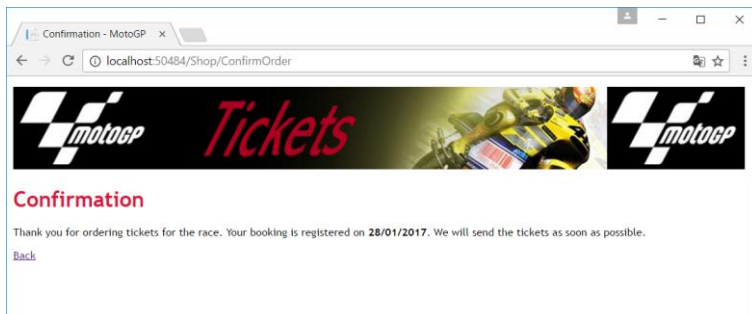
Maak nu de view aan. Dit is het resultaat:



Met *Back* ga je terug naar het menu.

6. Tickets bevestigen

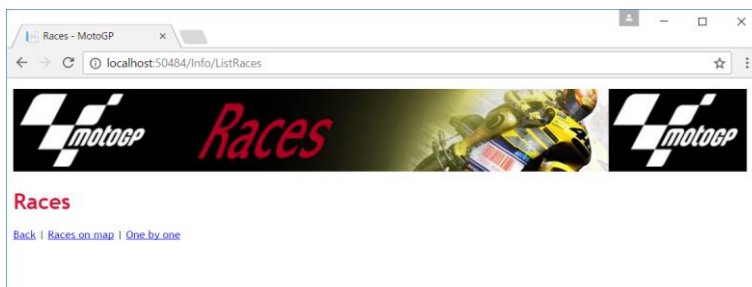
Maak in de *Shop-controller* een extra methode *ConfirmOrder*. Plaats *BannerNr* op 3 en roep de view op. Maak de view aan zodat je krijgt (voorlopig uittesten via de adresbalk):



Deze pagina wordt later gebruikt om een bevestiging te tonen wanneer een bezoeker tickets bestelt. Merk op dat in de tekst die afgebeeld wordt de huidige datum gebruikt wordt in het formaat *dd/mm/jjjj*. Als je op de hyperlink klikt, dan kom je terug op het hoofdmenu.

7. Races opvragen

Maak een extra *controller* *Info*. Maak hierin een methode *ListRaces* met bijhorende *view*. Het resultaat is voorlopig:



Zorg ervoor dat je de juiste *banner* (nummer 0) en *title* gebruikt. Onder de titel zie je drie hyperlinks. *Races on map* springt naar *Info/BuildMap* en *One by one* naar *Info/SelectRace*.

8. Race klasse

Maak in de *Models-map* een extra klasse *Race.cs*. Geef met de verkorte *CSharp* schrijfwijze deze klasse de volgende properties:

```
public class Race
```

```

{
    public int RaceID { get; set; }
    public string Name { get; set; }
    public int X { get; set; }
    public int Y { get; set; }
}

```

9. Races on map

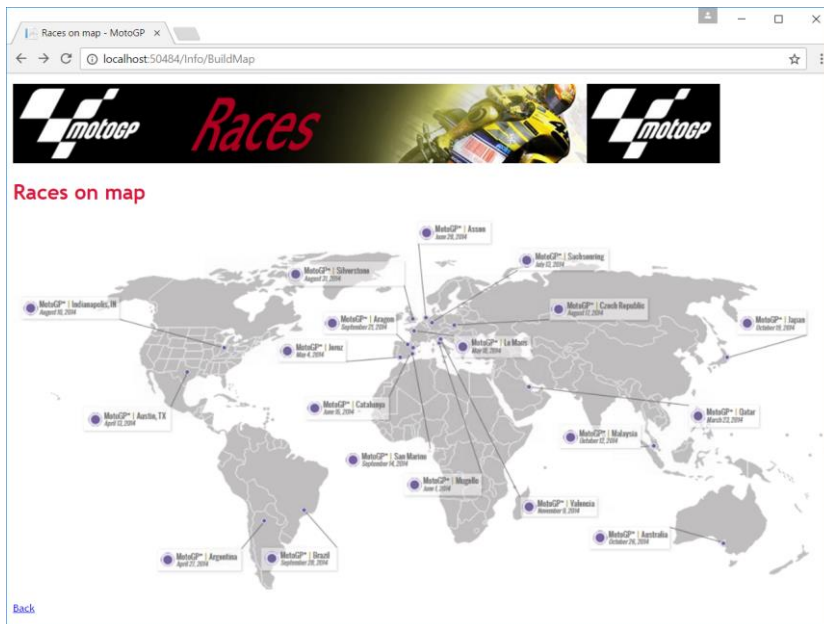
Maak in de *Info-controller* een extra methode *BuildMap* en bijhorende view. Vooraleer je de view oproept, maak je een *List* van *Race*-objecten en je stopt daarin (voorlopig) *hardcoded* volgende races:

```

{ RaceID = 1, X = 517, Y = 19, Name = "Assen" }
{ RaceID = 2, X = 859, Y = 249, Name = "Losail Circuit" }
{ RaceID = 3, X = 194, Y = 428, Name = "Autódromo Termas de Río Hondo" }

```

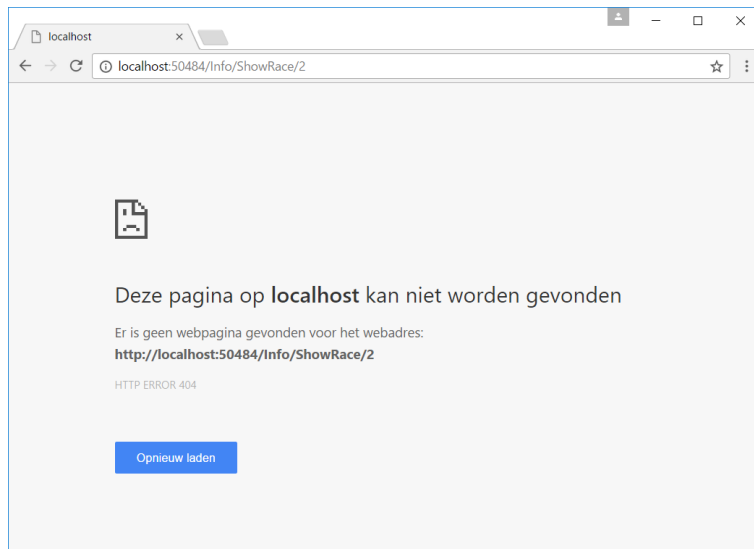
Geef die races (naast *BannerNr*) mee door naar de view. In die view toon je een afbeelding met alle MotoGP races in de wereld (*map.png*):



Maak die map nu *clickable* (meer info op http://www.w3schools.com/tags/tag_area.asp). Voor elke race uit de *List* (voorlopig zijn er dat drie) maak je een rondje waarop je kan klikken. Als je klikt op een race dan spring je naar de *ShowRace*-methode van de *Info-controller*. De *id* van de race is de eerste parameter (gebruik de *url helper*). Bijvoorbeeld voor *Losail Circuit* in Qatar:

/Info/ShowRace/2

Voorlopig krijg je volgende foutmelding omdat de methode *ShowRace* nog niet bestaat:



10. Test

Probeer nog eens vlug de hyperlinks uit vanaf de homepage:

