C# Web - Introductie

PRO - C# Web 1

DE HOGESCHOOL
MET HET NETWERK

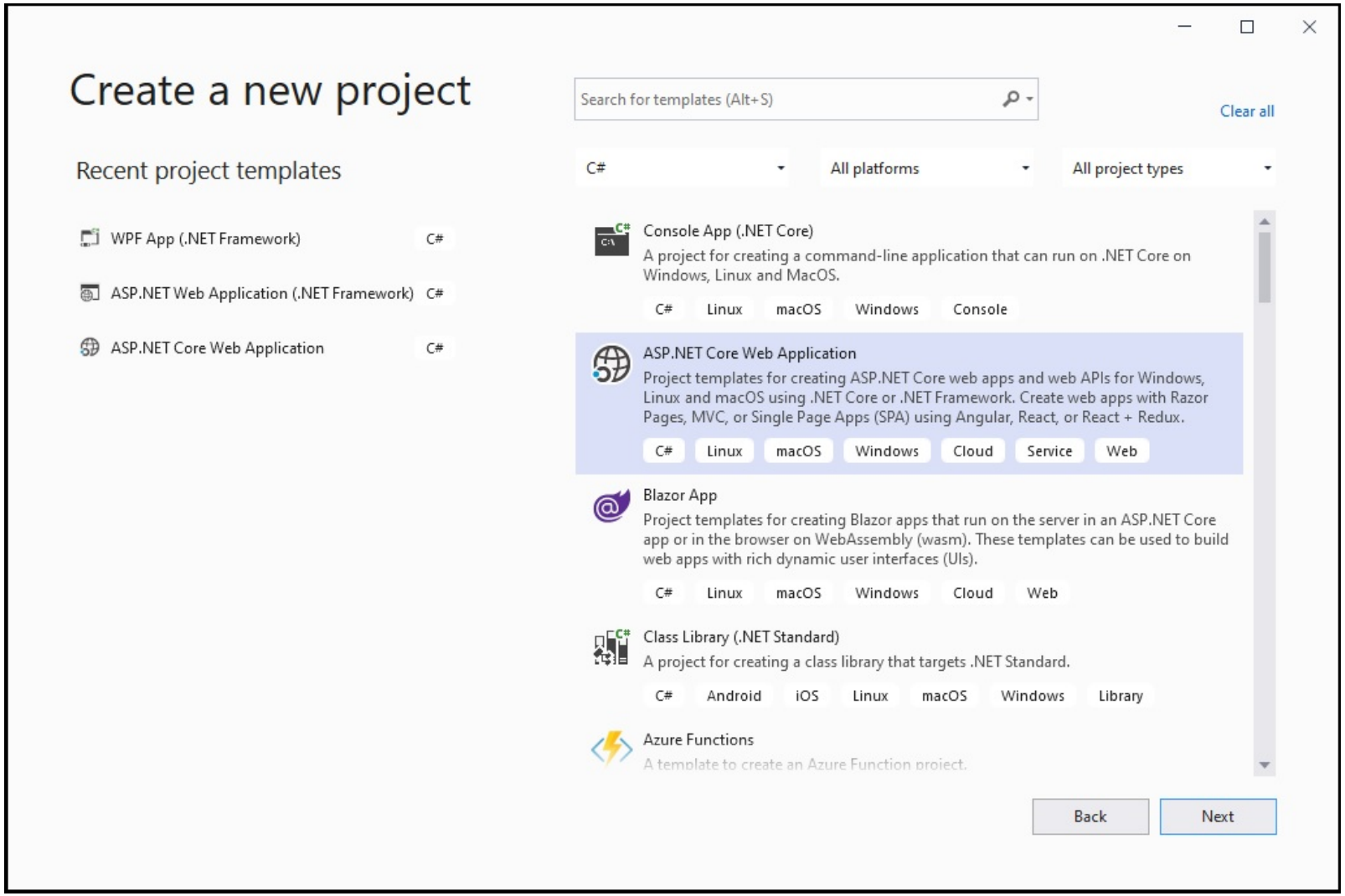
Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

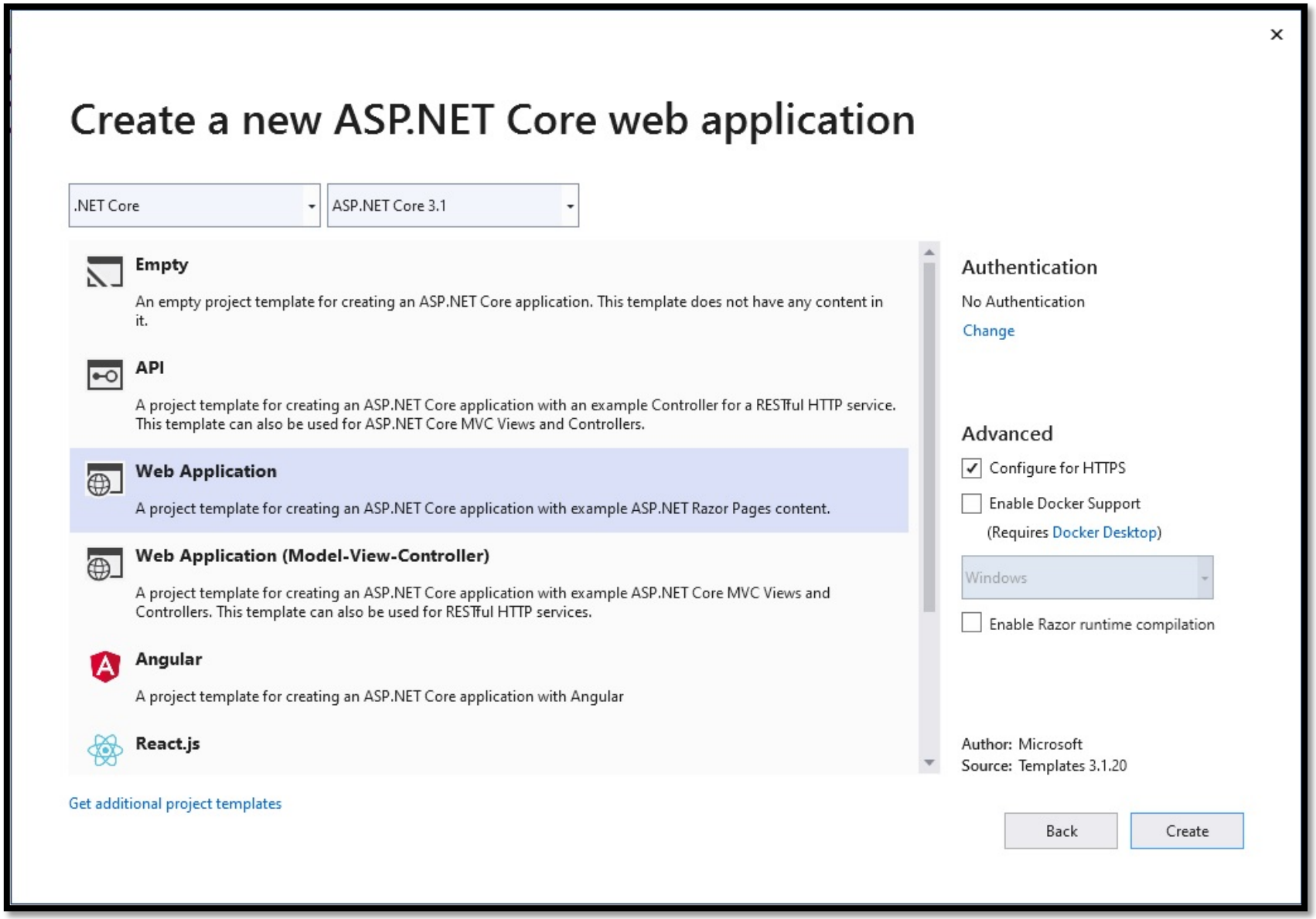
Doel

- ASP.Net Core toepassing
  - Razor Web Application

ASP.NET CORE

Razor Web Application

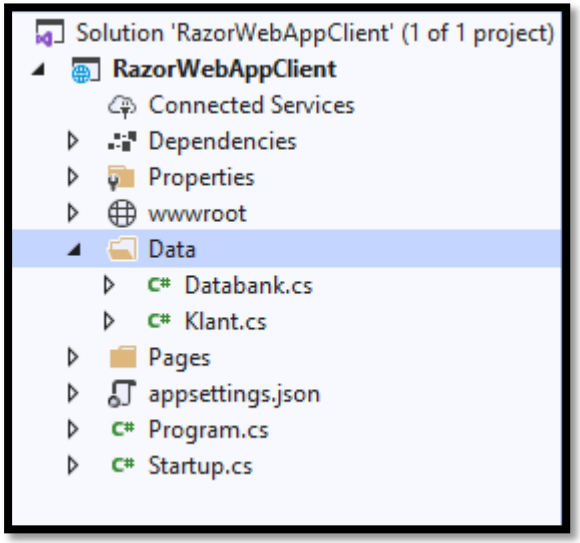

Create a new project
Search for templates (Alt+S)
Clear all
Recent project templates
WPF App (.NET Framework) C#
ASP.NET Web Application (.NET Framework) C#
ASP.NET Core Web Application C#
C#
All platforms
All project types
Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console
ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web
Blazor App
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web
Class Library (.NET Standard)
A project for creating a class library that targets .NET Standard.
C# Android iOS Linux macOS Windows Library
Azure Functions
A template to create an Azure Function project.
Back
Next

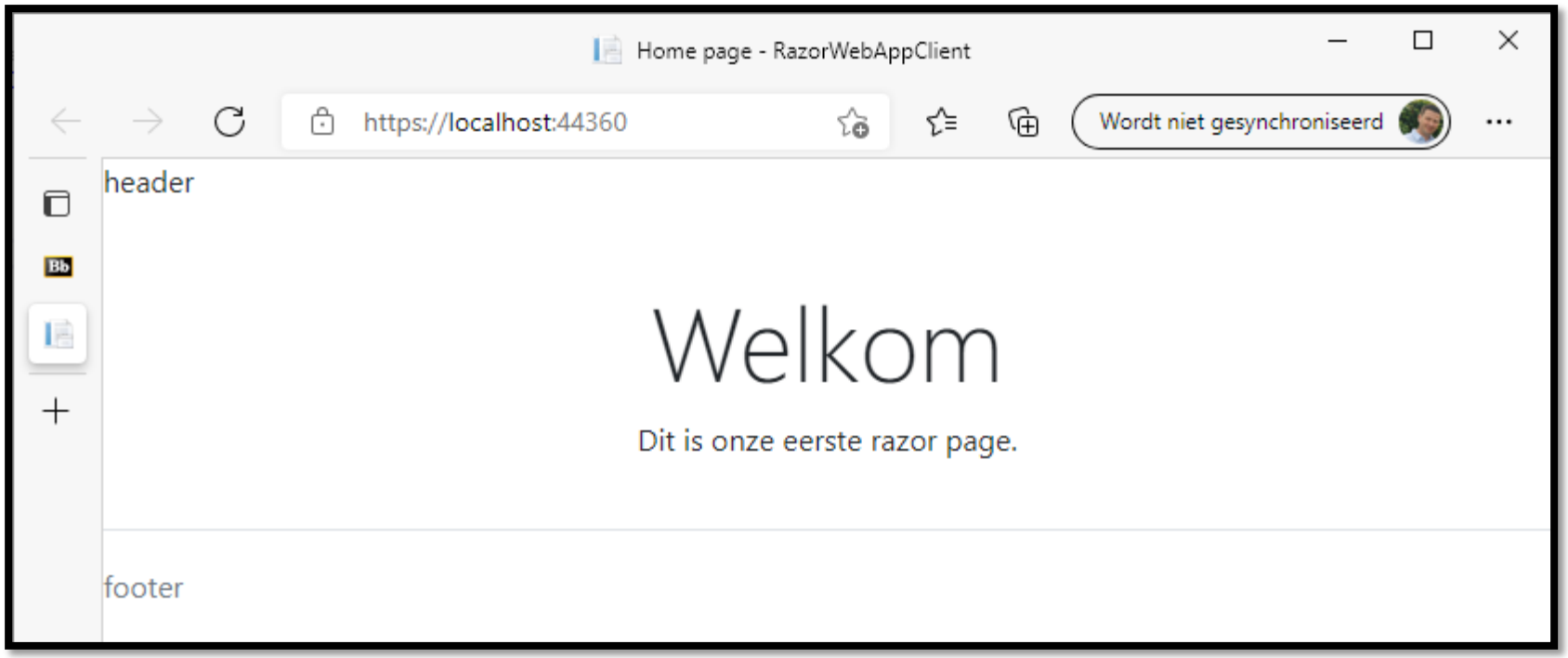

Create a new ASP.NET Core web application
.NET Core
ASP.NET Core 3.1
Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
Angular
A project template for creating an ASP.NET Core application with Angular
React.js
Authentication
No Authentication
Change
Advanced
Configure for HTTPS
Enable Docker Support
(Requires Docker Desktop)
Windows
Enable Razor runtime compilation
Author: Microsoft
Source: Templates 3.1.20
Get additional project templates
Back
Create

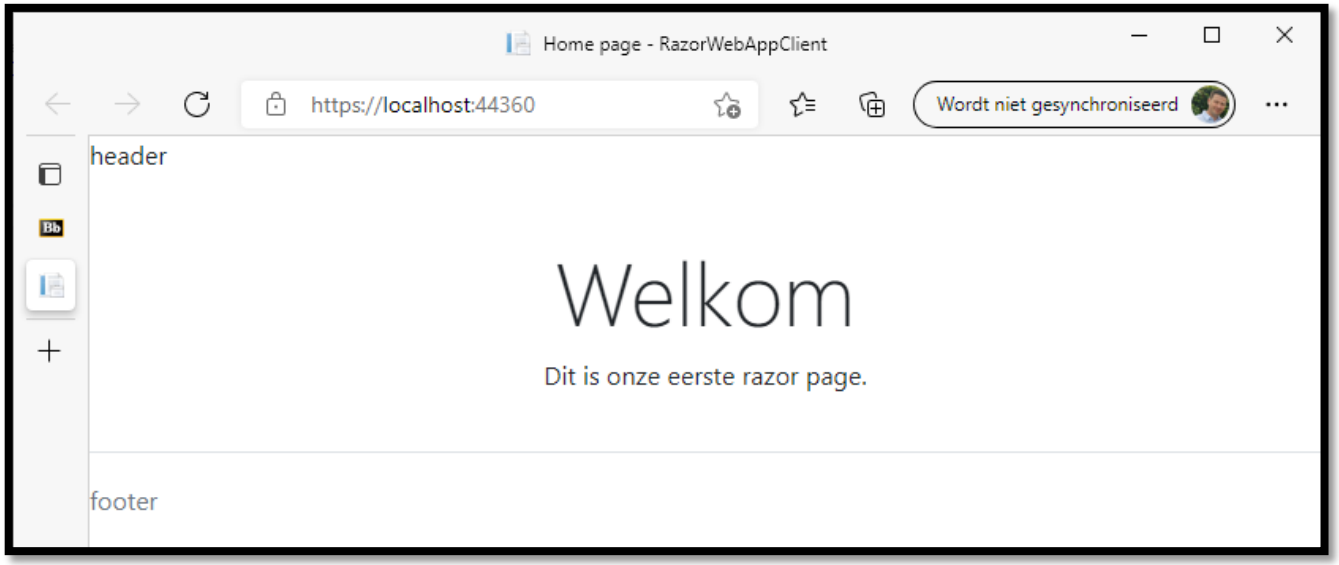
- Middleware is al geactiveerd voor Razor content

```
public void ConfigureServices(IServiceCollection services)
{
            services.AddRazorPages();
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
          …
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapRazorPages();
            });
}
```

- Project instellingen
  - Add Folder Data
    - Add class
      - Klant.cs
      - Databank.cs

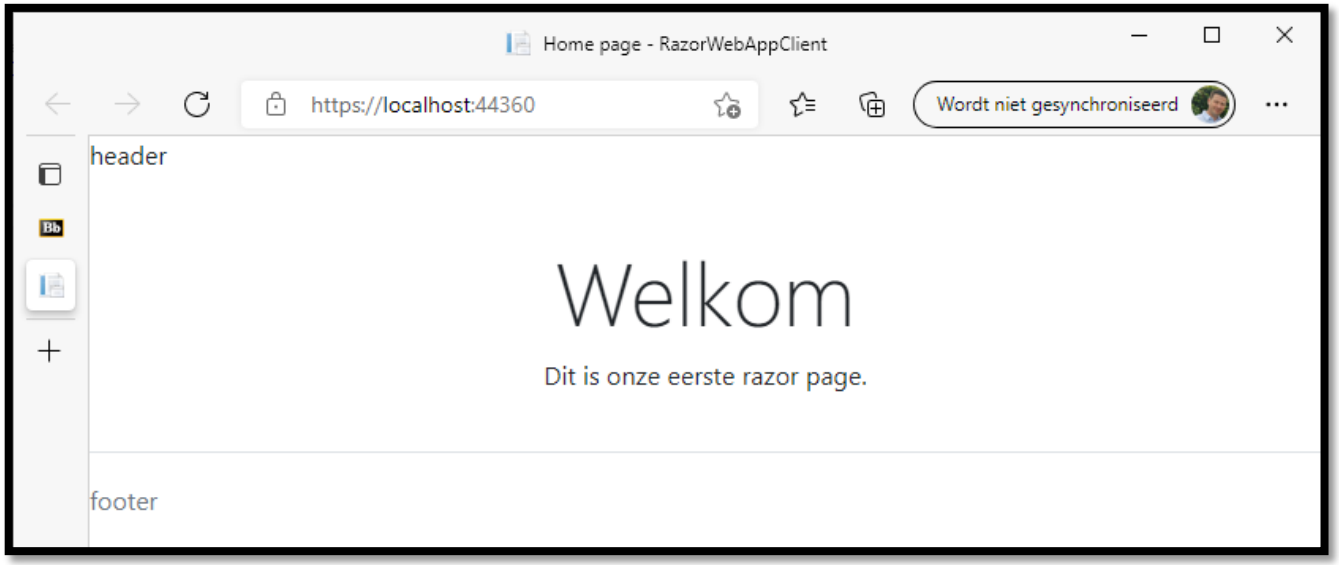



Home page - RazorWebAppClient
https://localhost:44360
Wordt niet gesynchroniseerd
header
Welkom
Dit is onze eerste razor page.
footer



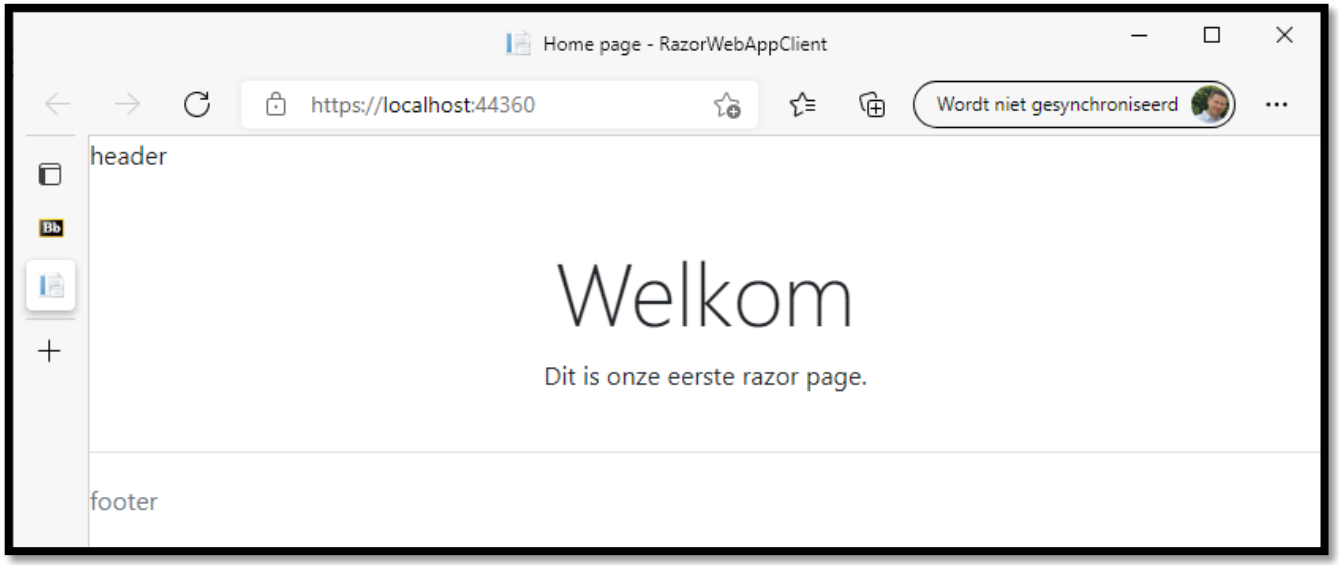
Pages/Index.cshtml

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welkom</h1>
    <p>Dit is onze eerste razor page.</p>
</div>
```



Pages/Index.cshtml

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welkom</h1>
    <p>Dit is onze eerste razor page.</p>
</div>
```

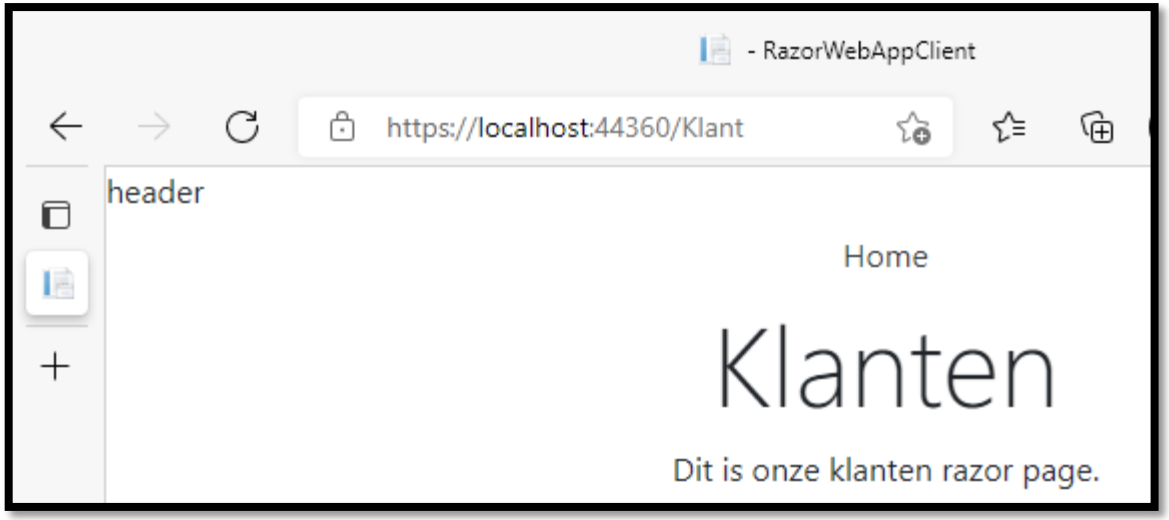


Pages/Shared/_Layout.cshtml

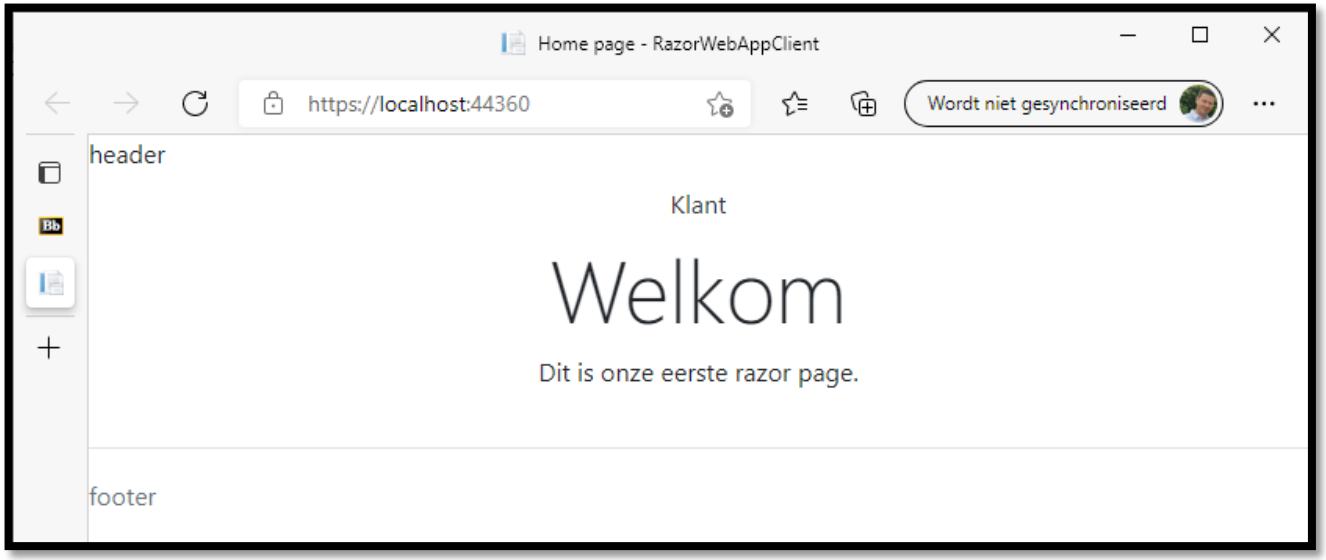
```
<body>
    <header>
        header
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>
    <footer class="border-top footer text-muted">
        footer
    </footer>
```

Project instellingen

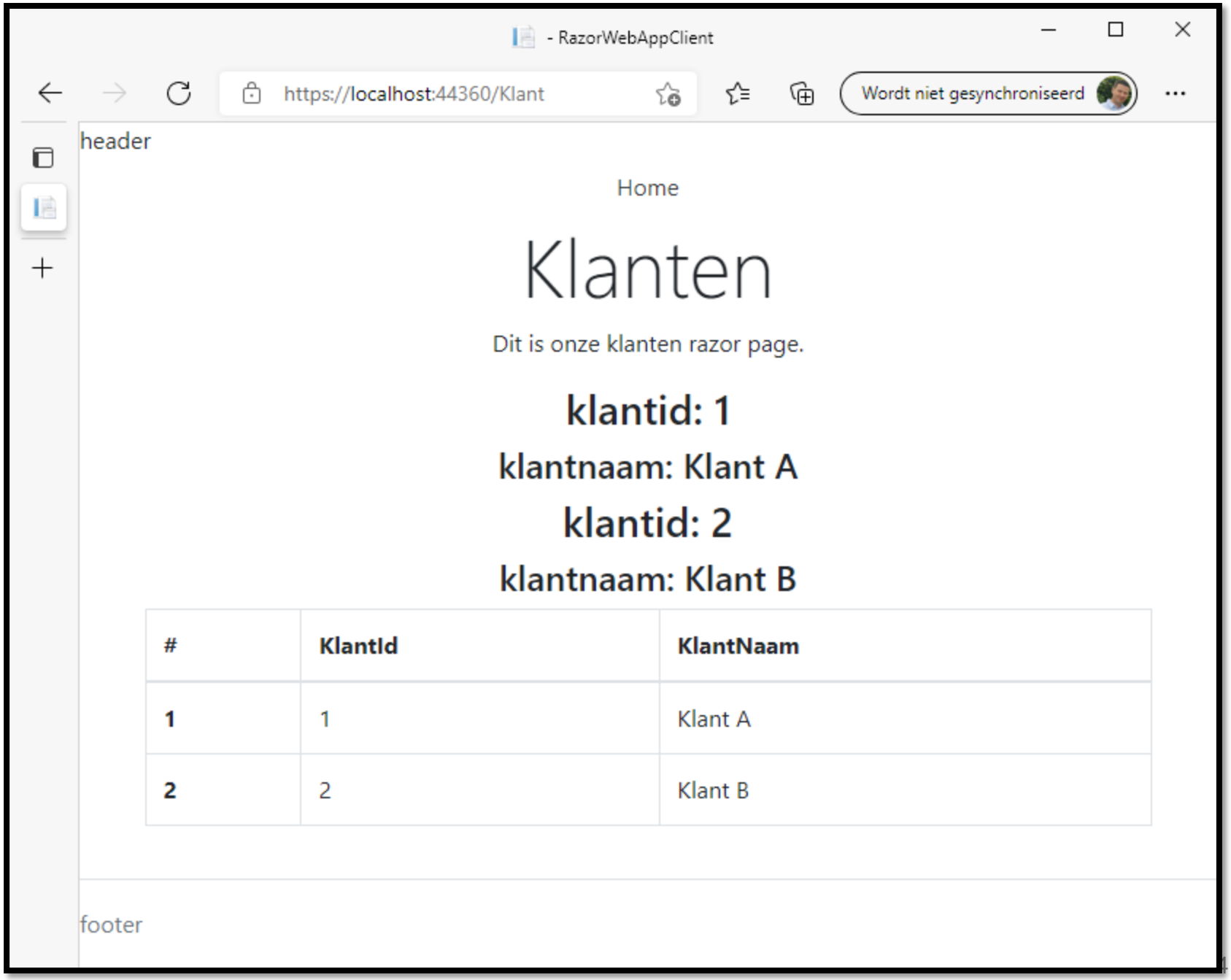
- Pages – Add Page
  - Klant.cshtml

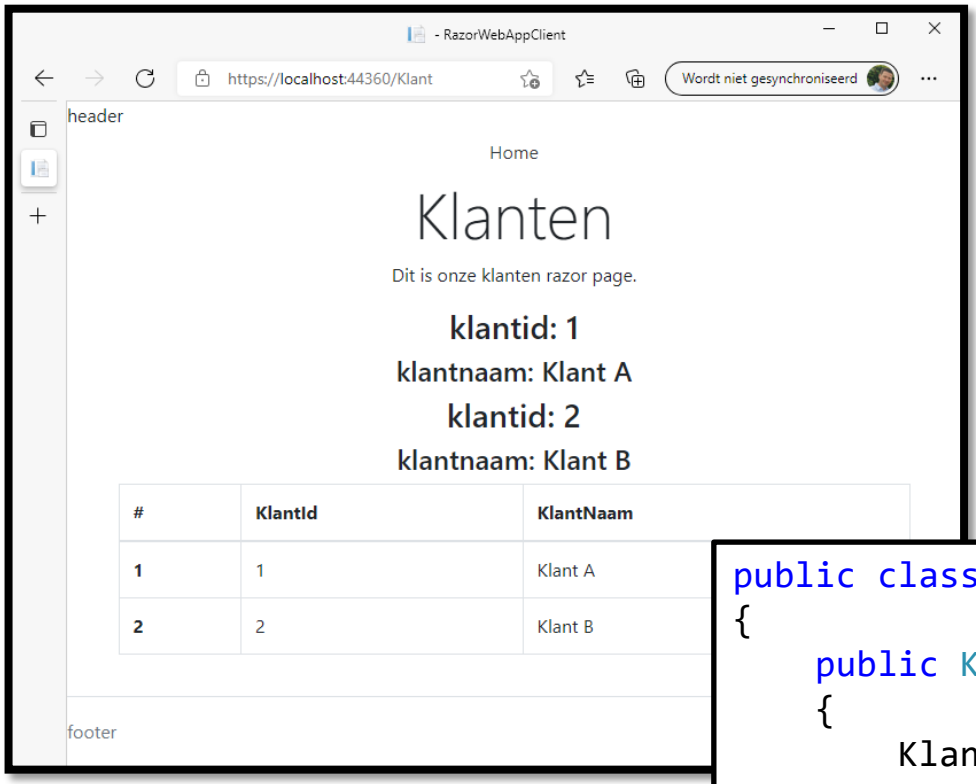



```
@page
@model RazorWebAppClient.Pages.KlantModel
@{
}
<div class="text-center">
    <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
    <h1 class="display-4">Klanten</h1>
    <p>Dit is onze klanten razor page.</p>
```

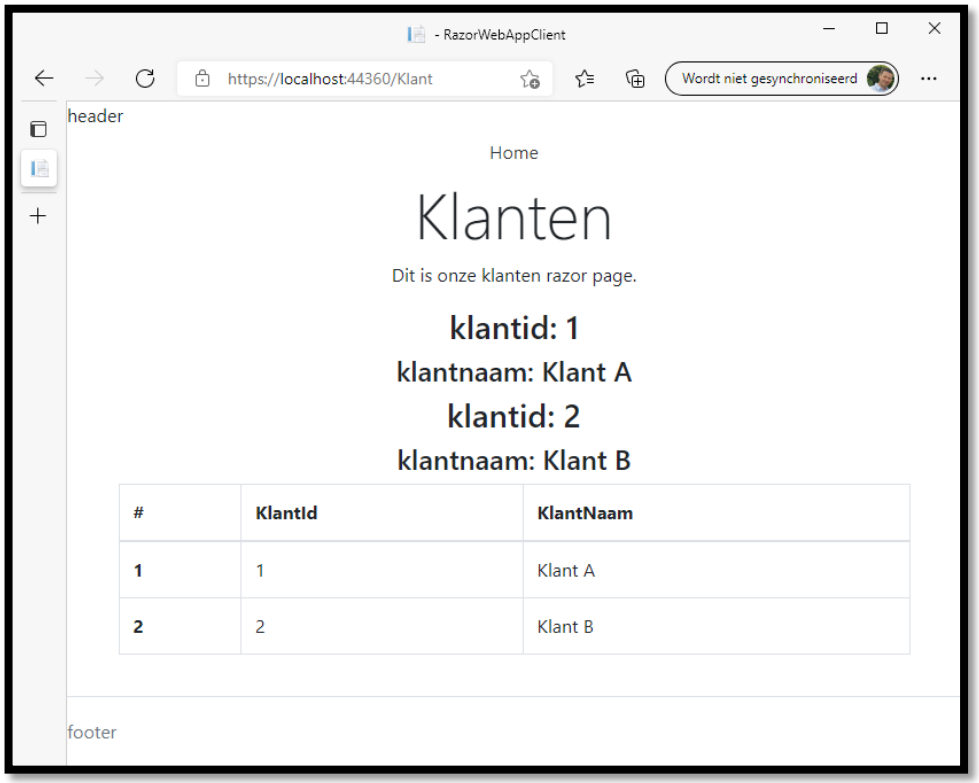



```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <a class="nav-link text-dark" asp-area="" asp-page="/Klant">Klant</a>
    <h1 class="display-4">Welkom</h1>
    <p>Dit is onze eerste razor page.</p>
</div>
```


- RazorWebAppClient
https://localhost:44360/Klant
Wordt niet gesynchroniseerd
header
Home
Klanten
Dit is onze klanten razor page.
klantid: 1
klantnaam: Klant A
klantid: 2
klantnaam: Klant B
KlantId KlantNaam
1 1 Klant A
2 2 Klant B
footer




```
public class Klant
{
    public Klant()
    {
        KlantId = -1;
        KlantNaam = string.Empty;
    }
    public Klant(int id, string naam)
    {
        KlantId = id;
        KlantNaam = naam;
    }
    public int KlantId { get; set; }
    public string KlantNaam { get; set; }
    public bool GevalideerdeKlant => (KlantId > -1);
}
```




```
public class Databank
{
    public static List<Klant> Klanten = new List<Klant>();
    public static void StartDatabank()
    {
        Klanten.Add(new Klant(1, "Klant A"));
        Klanten.Add(new Klant(2, "Klant B"));
    }
}
```

ProjectInstellingen

- Startup.cs
- Starten van de statische databank.

```
using RazorWebAppClient.Data;
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    ...
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapRazorPages();
    });
    Databank.StartDatabank();
}
```

Pages/Klant.cshtml

```
@page
@model RazorWebAppClient.Pages.KlantModel
@{
    var klanten = RazorWebAppClient.Data.Databank.Klanten;
    var rowId = 0;
}
<div class="text-center">
    <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
    <h1 class="display-4">Klanten</h1>
    <p>Dit is onze klanten razor page.</p>
    @foreach (var klant in klanten)
    {
        <div>
            <h3>klantid: @klant.KlantId</h3>
            <h4>klantnaam: @klant.KlantNaam</h4>
        </div>
    }
</div>
```

Pages/Klant.cshtml

```
…
<table class="table table-bordered">
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">KlantId</th>
            <th scope="col">KlantNaam</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var klant in klanten)
        {
            rowId++;
            <tr>
                <th scope="row">@rowId</th>
                <td>@klant.KlantId</td>
                <td>@klant.KlantNaam</td>
            </tr>
        }
    </tbody>
</table>
```

Pages/Klant.cshtml

```
…
<table class="table table-bordered">
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">KlantId</th>
            <th scope="col">KlantNaam</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var klant in klanten)
        {
            rowId++;
            <tr>
                <th scope="row">@rowId</th>
                <td>@klant.KlantId</td>
                <td>@klant.KlantNaam</td>
            </tr>
        }
    </tbody>
</table>
```

ASP.NET CORE

Razor Web Application – Oefening01

- Project instellingen
  - Add class
    - Location.cs
      - LocationID
      - Postcode
      - City
    - Databank.cs
      - Voeg 2 records toe
        - 1-3500-Hasselt
        - 2-3600-Genk
  - Add razor Page
    - Locaties.cshtml
      - Toon een Tabel met de gegevens van de locaties uit de databank class.
  - Voeg een hyperlink toe in de Index razor page naar de nieuwe pagina

```
public class Location
{
    public Location()
    {
        LocationId = -1;
        PostCode = string.Empty;
        City = string.Empty;
    }
    public Location(int locationId, string postCode, string city)
    {
        LocationId = locationId;
        PostCode = postCode;
        City = city;
    }
    public int LocationId { get; set; }
    public string PostCode { get; set; }
    public string City { get; set; }
}
```