



C# Web – MVC

PRO/PRW - 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
 - MVC Web Application
 - Models
 - Views
 - Controllers
 - Entity framework
 - Oefening 08
 - Oefening 09

ASP.NET CORE

MVC Web Application

- Oefening 08

Oefening 08 – 01. Nieuw Project



ASP.NET Core Web App (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.

C#

Linux

macOS

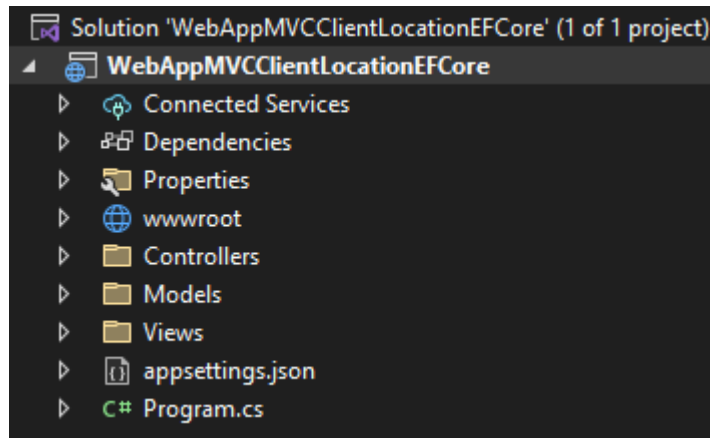
Windows

Cloud

Service

Web

WebAppMVCClietLocationEFCore



Oefening 08 – 02. NuGet – SQL Server

NuGet - Solution WebAppMVCCli...re: Overview

Browse Installed Updates Consolidate Manage Packages for Solution

sqlserver x ↻ ☐ Include prerelease Package source: nuget.org

Microsoft.EntityFrameworkCore.SqlServer
Microsoft SQL Server database provider for Entity Framework Core.

Microsoft.Extensions.Caching.SqlServer by
Distributed cache implementation of Microsoft.Extensions.Caching.Distributed.I...

Microsoft.SqlServer.Server by Microsoft, 41,7M
This is a helper library for Microsoft.Data.SqlClient enabling cross fram...

Hangfire.SqlServer by Sergey Odinkov, 55,9M d
SQL Server 2008+ (including Express), SQL Server LocalDB and SQL Azure storage suppo...

AspNetCore.HealthChecks.SqlServer by Xabaril
HealthChecks.SqlServer is the health check

Microsoft.EntityFrameworkCore.SqlServer nuget.org

Versions - 0

<input checked="" type="checkbox"/>	Project
<input checked="" type="checkbox"/>	WebAppMVCCClientLocationEFCore

Installed: not installed Uninstall

Version: 6.0.22 Install

Solution 'WebAppMVCCClientLocationEFCore' (1 of 1 project)

WebAppMVCCClientLocationEFCore

- Connected Services
- Dependencies
 - Analyzers
 - Frameworks
 - Packages
 - Microsoft.EntityFrameworkCore.SqlServer (6.0.22)
- Properties
- wwwroot
- Controllers
- Models
- Views
- appsettings.json
- C# Program.cs

Oefening 08 – 03. Maak een database class aan

- Create Folder
 - Data
 - Add class
 - ClientLocationContext.cs
 - Inherit baseClass DbContext

```
using Microsoft.EntityFrameworkCore;

namespace WebAppMVCCClientLocationEFCore.Data
{
    public class ClientLocationContext : DbContext
    {
    }
}
```

Oefening 08 – 03. Maak een database class aan

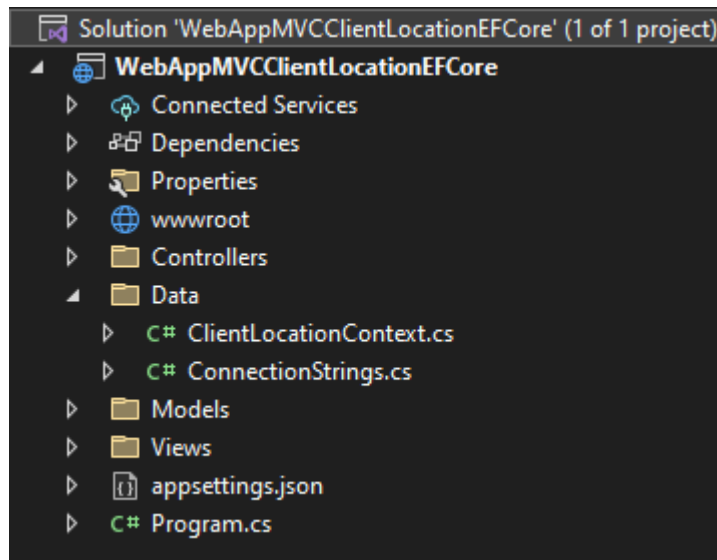
```
using Microsoft.EntityFrameworkCore;

namespace WebAppMVCCClientLocationEFCore.Data
{
    public class ClientLocationContext : DbContext
    {
        public ClientLocationContext(
            DbContextOptions<ClientLocationContext> options) :
            base(options)
        { }
    }
}
```

Oefening 08 – 04. Voeg een connection string toe

- Data folder – Add class – Connectionstrings – Add constant LocalDb

```
namespace WebAppMVCCClientLocationEFCore.Data
{
    public class ConnectionStrings
    {
        public const string LocalDb = @"Server=(localdb)\MSSQLLocalDB;
            Database=ClientLocationData; MultipleActiveResultSets=true";
    }
}
```



Oefening 08 – 05. Voeg de database service toe in je program.cs class

```
using Microsoft.EntityFrameworkCore;
using WebAppMVCCClientLocationEFCore.Data;

var builder = WebApplication.CreateBuilder(args);

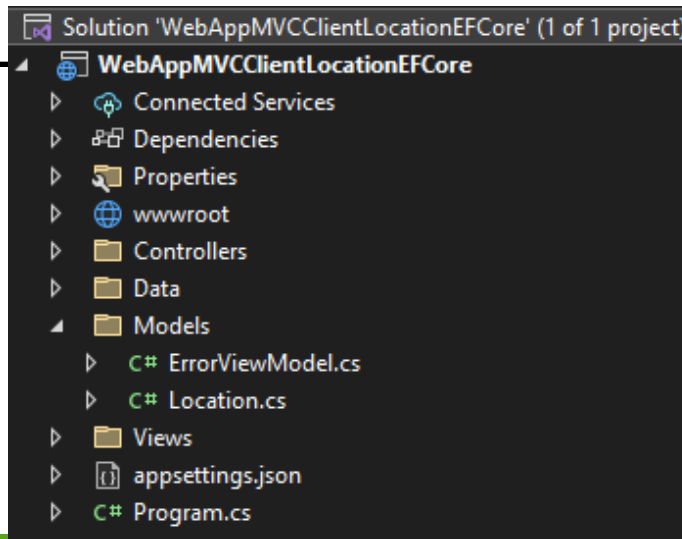
var connectionString = ConnectionStrings.LocalDb;

builder.Services.AddDbContext<ClientLocationContext>(
    options => options.UseSqlServer(connectionString));

builder.Services.AddControllersWithViews();

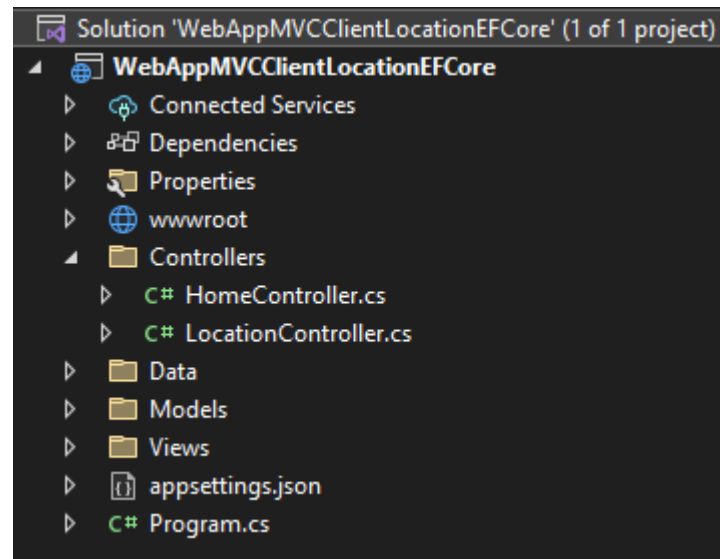
var app = builder.Build();
```

```
namespace WebAppMVCCClientLocationEFCore.Models
{
    public class Location
    {
        public int LocationId { get; set; }
        [StringLength(15)]
        [Required]
        public string PostCode { get; set; }
        [StringLength(100)]
        [Required]
        public string City { get; set; }
    }
}
```



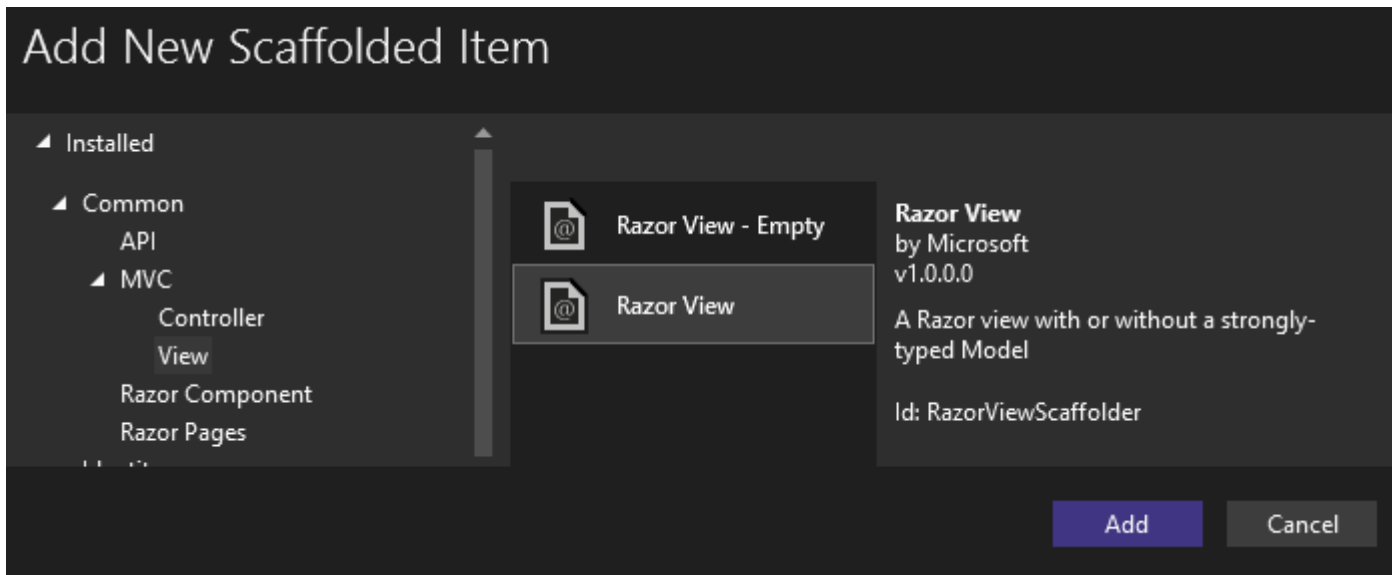
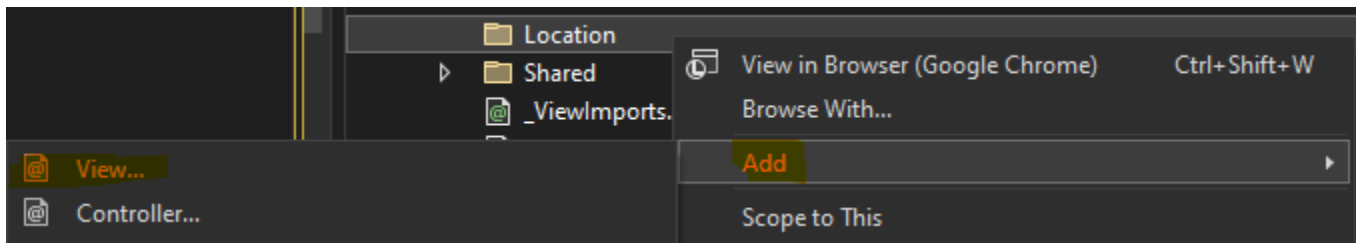
```
namespace WebAppMVCCClientLocationEFCore.Controllers
{
    public class LocationController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
        [HttpGet]
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Create(Location location)
        {
            return View();
        }
    }
}
```

```
namespace WebAppMVCClietLocationEFCore.Controllers
{
    public class LocationController : Controller
    {
        ClientLocationContext _context;
        public LocationController(ClientLocationContext context)
        {
            _context = context;
        }
        ...
    }
}
```



Oefening 08 – 12a. Views – Voeg de views toe

- Create Folder (in Folder Views)
 - Location (Views/Location)
 - Add Razor View (List template)
 - Index.cshtml



Oefening 08 – 12a. Views – Voeg de views toe

- Create Folder (in Folder Views)
 - Location
 - Add Razor View (List template)
 - Index.cshtml

Add Razor View

View name:

Template:

Model class:

DbContext class:

Options

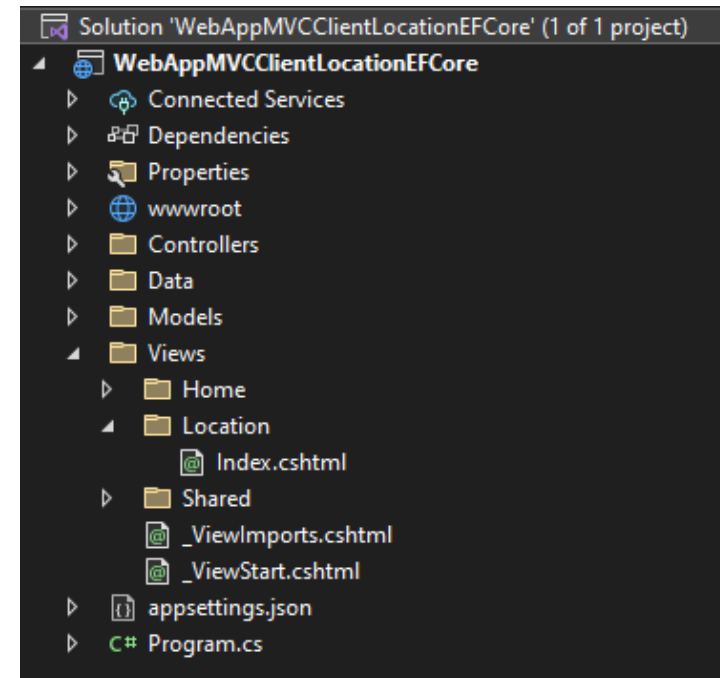
- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page

...

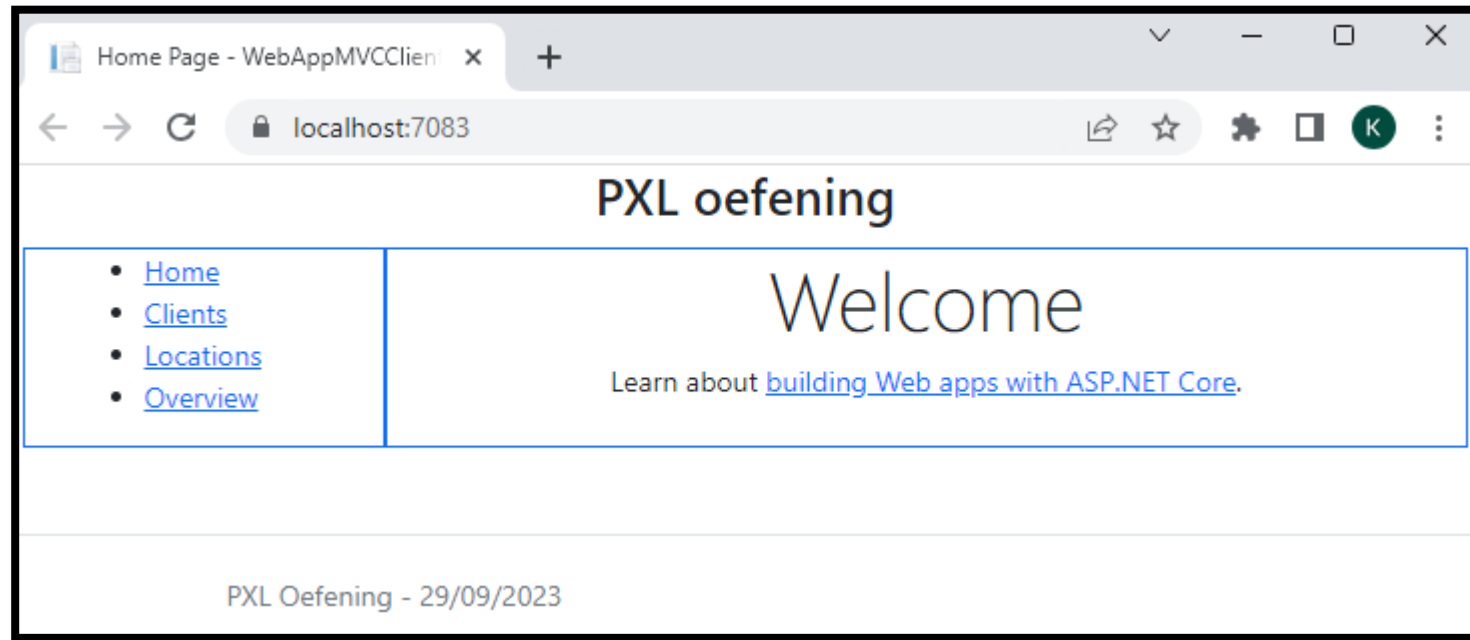
(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel



Oefening 08 – 16. Layout – Styling the content



Oefening 08 – 16. Layout – Styling the content

```
<header>
  <div class="text-center">
    <h2>PXL oefening</h2>
  </div>
</header>
<div class="row m-1 p-1">
  <div id="menu" class="col-3 border border-primary">
    <div class="container">
      <ul>
        <li><a asp-controller="Home" asp-action="Index">Home</a></li>
        <li><a asp-controller="" asp-action="">Clients</a></li>
        <li><a asp-controller="Location" asp-action="Index">Locations</a></li>
        <li><a asp-controller="" asp-action="">Overview</a></li>
      </ul>
    </div>
  </div>
  <div id="main" class="col-9 border border-primary">
    @RenderBody()
  </div>
</div>
<footer class="border-top footer text-muted">
  <div class="container">
    PXL Oefening - @DateTime.Now.ToString("dd/MM/yyyy")
  </div>
</footer>
```


ASP.NET CORE

MVC Web Application

- Oefening 09