



# C# Web - Introductie

## PRO - C# Web 1

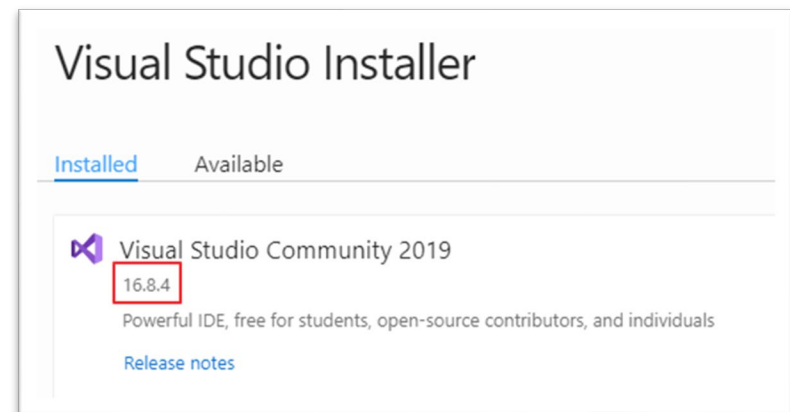
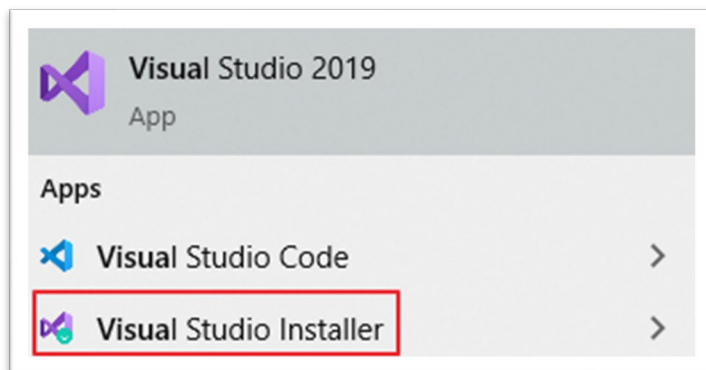
### DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.pxl.be/facebook)



# Benodigde software

- Visual Studio 2019
  - Versie 16.8.4 of hoger
  - Start het programma “Visual Studio Installer” en controleer het versienummer. Klik op de “Update” knop als de versie te laag is



# Benodigde software

Modifying — Visual Studio Enterprise 2019 — 16.8.5

## Workloads

## Individual components

## Language packs

## Installation locations

### Web & Cloud (4)



#### ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.



#### Python development

Editing, debugging, interactive development and source control for Python.



#### Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET



#### Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



### Desktop & Mobile (5)



#### .NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET



#### Desktop development with C++

Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.



#### Universal Windows Platform development

Create applications for the Universal Windows Platform with C#, VB, or optionally C++.



#### Mobile development with .NET

Build cross-platform applications for iOS, Android or Windows using Xamarin.

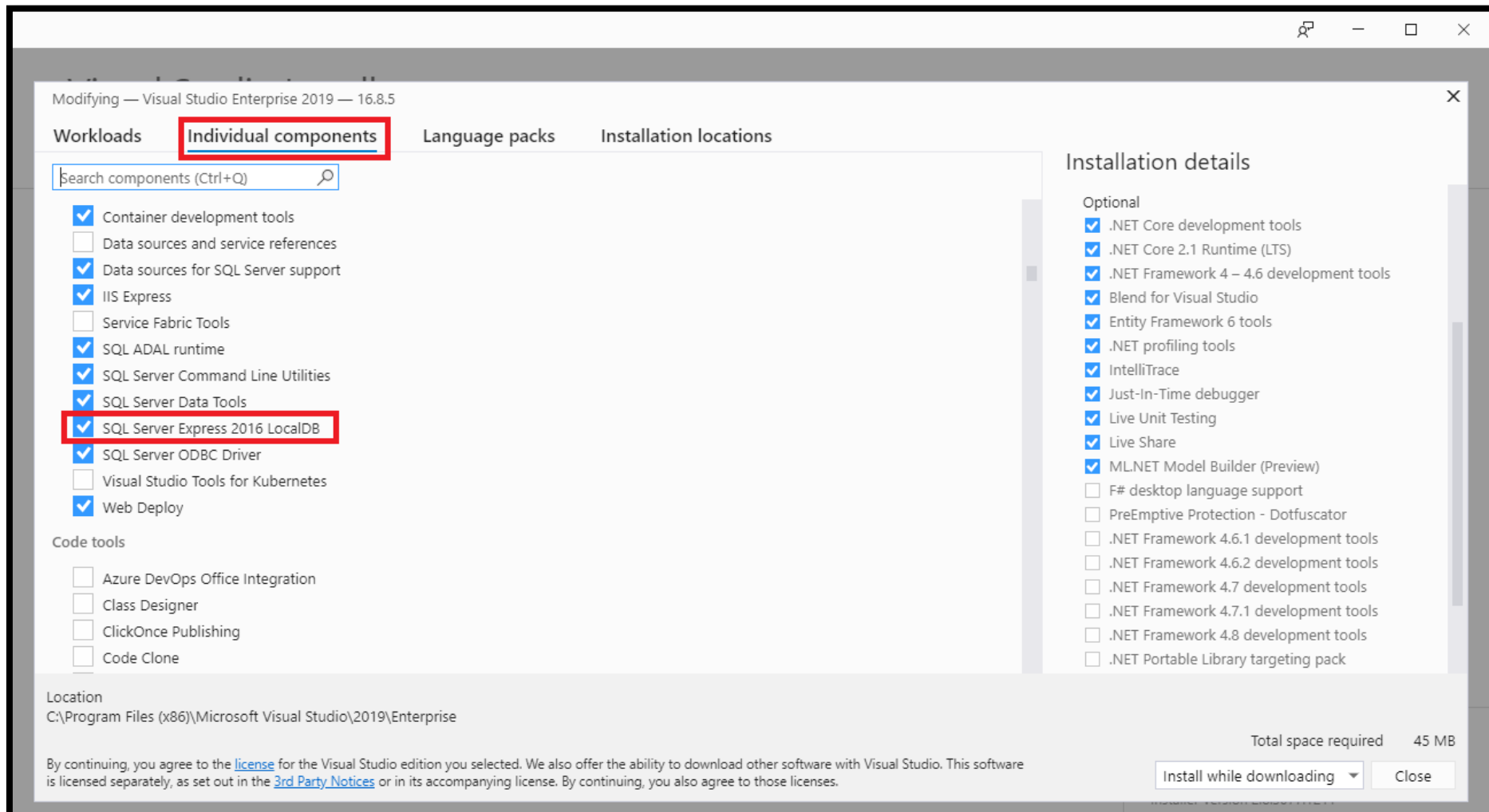


## Installation details

### Optional

- ☒ .NET Core development tools
- ☒ .NET Core 2.1 Runtime (LTS)
- ☒ .NET Framework 4 – 4.6 development tools
- ☒ Blend for Visual Studio
- ☒ Entity Framework 6 tools
- ☒ .NET profiling tools
- ☒ IntelliTrace
- ☒ Just-In-Time debugger
- ☒ Live Unit Testing
- ☒ Live Share
- ☒ ML.NET Model Builder (Preview)
- ☐ F# desktop language support
- ☐ PreEmptive Protection - Dotfuscator
- ☐ .NET Framework 4.6.1 development tools
- ☐ .NET Framework 4.6.2 development tools
- ☐ .NET Framework 4.7 development tools
- ☐ .NET Framework 4.7.1 development tools
- ☐ .NET Framework 4.8 development tools
- ☐ .NET Portable Library targeting pack

# Benodigde software



## Overzicht C# Web

- Empty Web Application
- Razor Web Application
- MVC concept
  - Model
    - Model Validation
  - View
  - Controller
- Entity Tools
- Entity Framework
- TagHelper
- View Component
- Identity
  - Users
  - Roles
- Built-in services
  - `services.AddControllersWithViews`
  - `Services.AddDbContext`
- Middleware componenten
  - `app.UseEndpoints`
  - `App.UseRouting`

## Evaluatie C# Web

- 80% open boek
- 20% project
  - Start week 3
  - Online 2u per week tem week7
  - 2 evaluatie momenten
    - Na 7 weken
    - Na Semester 1
  - 10/20 bij volgen richtlijnen
    - Extra punten te verdienen
    - Style, error handling, ...

# Doel

- Visual Studio installatie
- Een eerste ASP.Net Core toepassing aanmaken
  - Empty Web Application
  - Razor Web Application

# ASP.NET CORE

Empty Web Application



# Create a new project

## Recent project templates

-  WPF App (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) C#
-  ASP.NET Core Web Application C#

Search for templates (Alt+S)



[Clear all](#)

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

[Get additional project templates](#)

## Authentication

No Authentication

[Change](#)

## Advanced

☒ Configure for HTTPS

☐ Enable Docker Support  
(Requires [Docker Desktop](#))

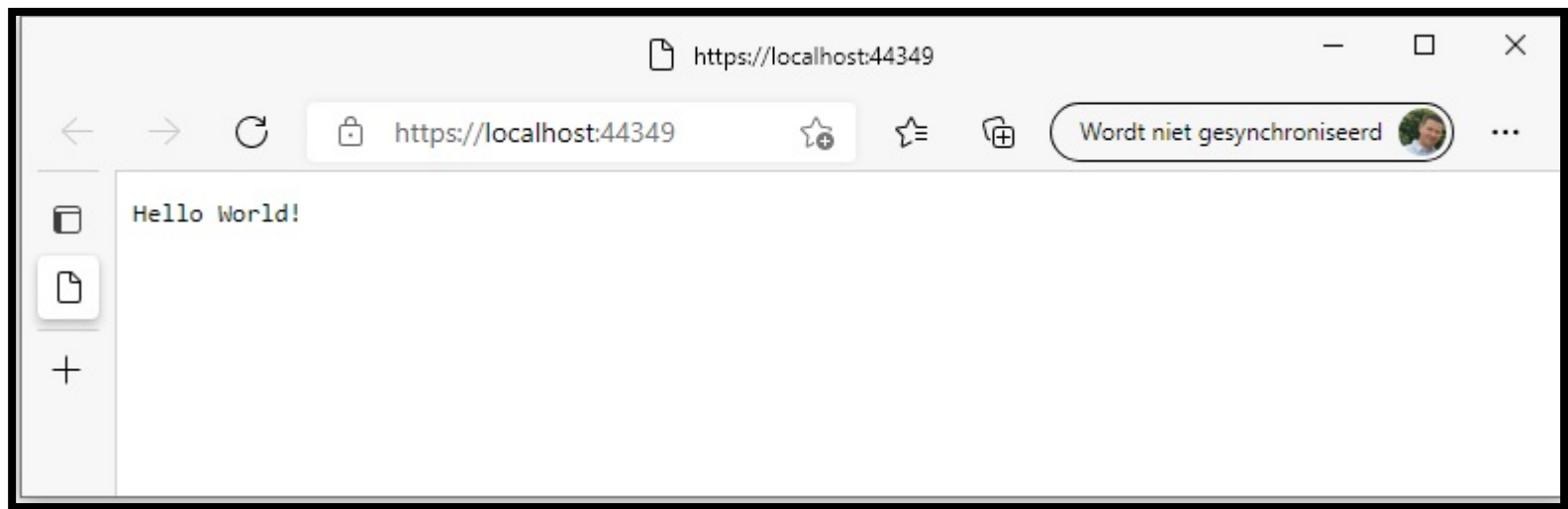
Windows

Author: Microsoft

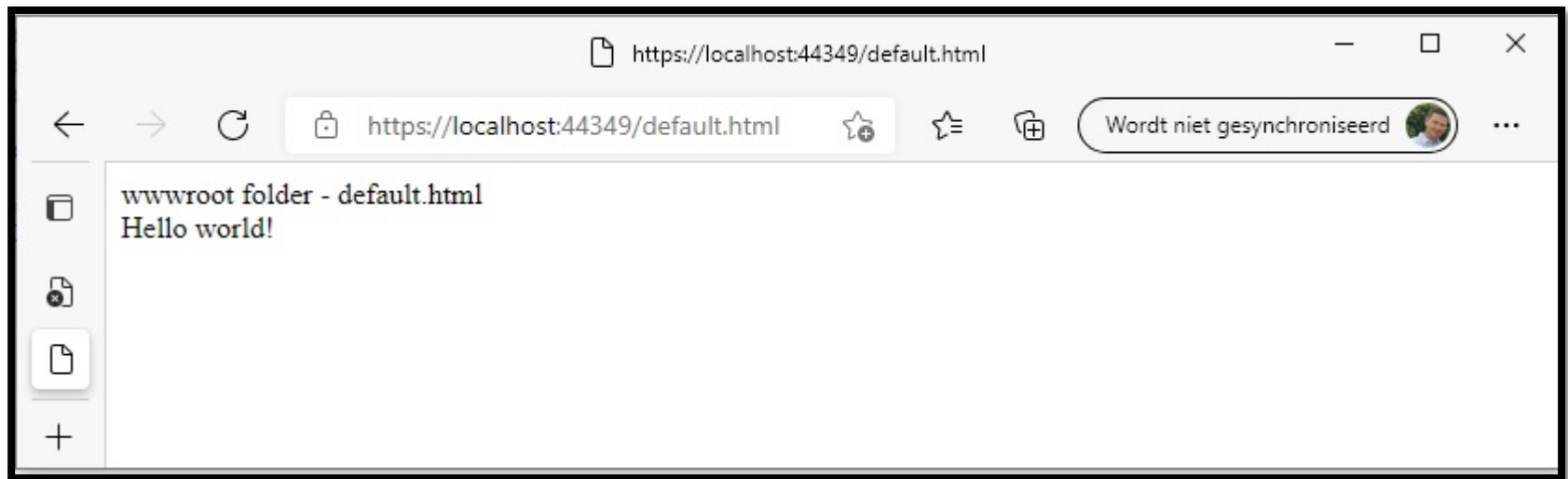
Source: Templates 3.1.20

Back

Create



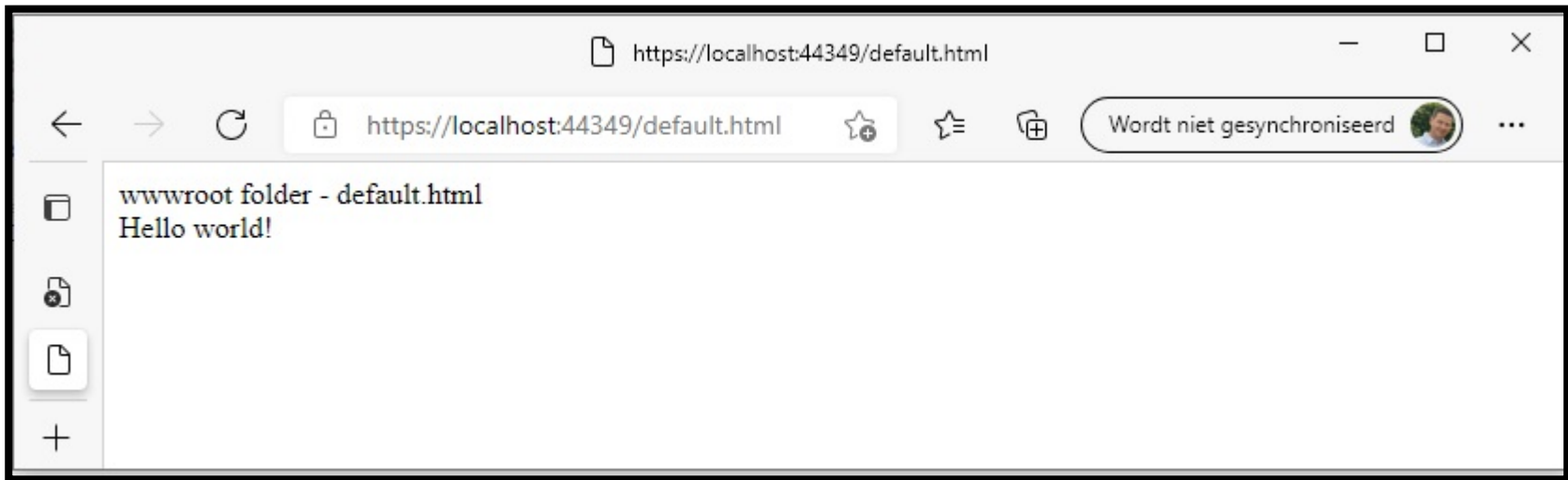
- Activeer middleware voor HTML content



- Project instellingen
  - Add Folder
    - wwwroot
      - Add HTML file
        - default.html

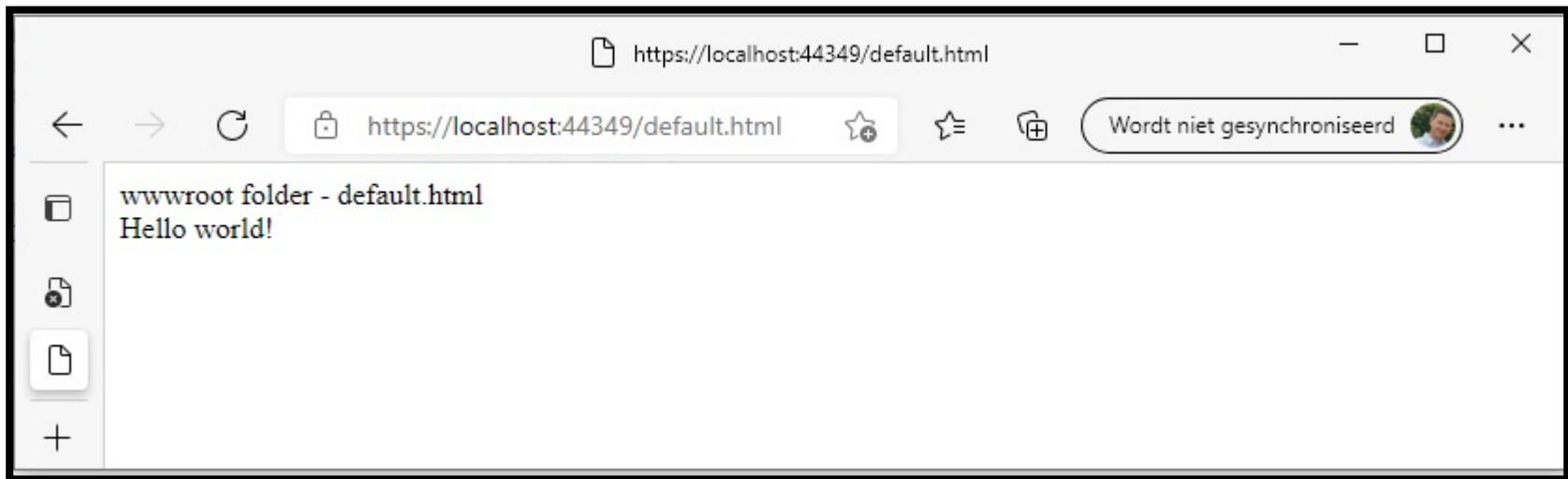
```
//endpoints.MapGet("/", async context =>  
    //{  
        //      await  
context.Response.WriteAsync("Hello World!");  
    //});
```

- Activeer middleware voor HTML content
  - Startup.cs
    - app.UseDefaultFiles();
    - app.UseStaticFiles();

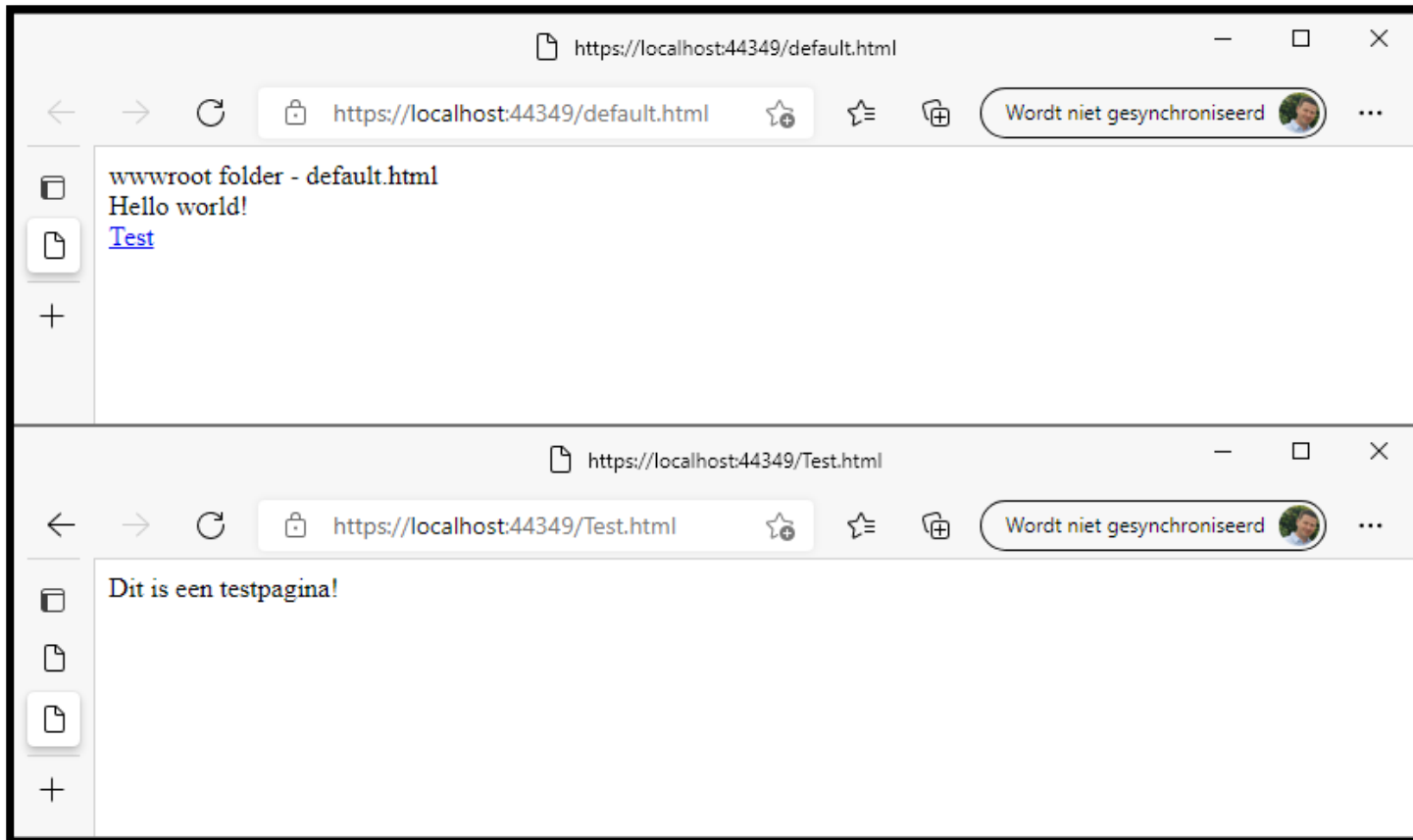


- Project instellingen
  - Add Folder
    - wwwroot
      - Add HTML file
        - default.html
- Activeer middleware voor HTML content
  - Startup.cs
    - `app.UseDefaultFiles();`
    - `app.UseStaticFiles();`

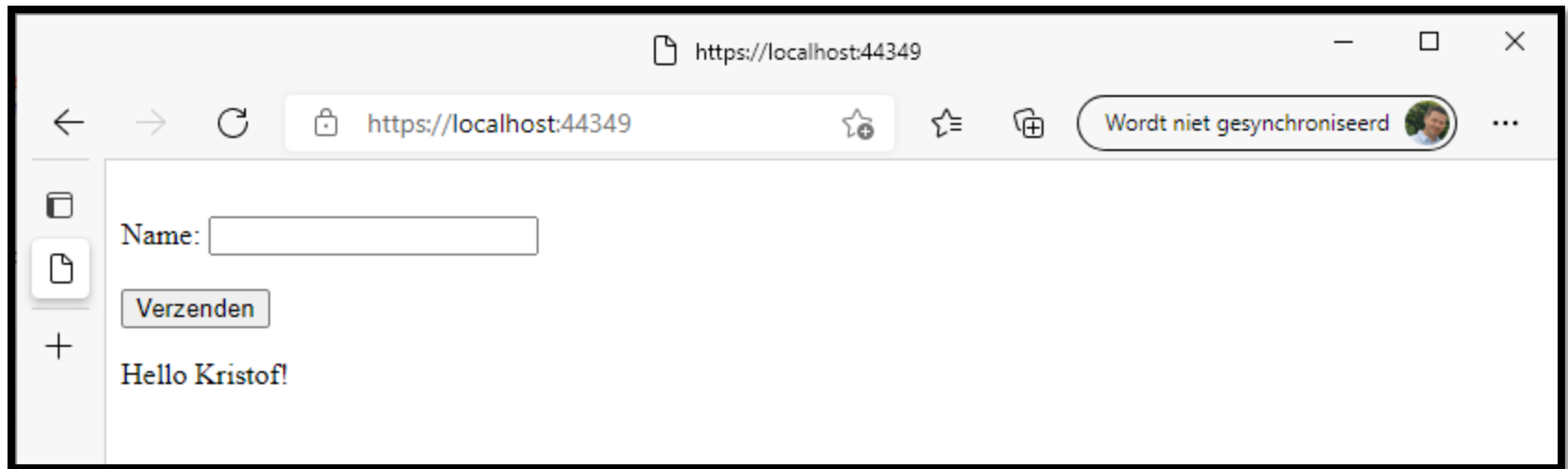
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
  wwwroot folder - default.html
  <br />
  Hello world!
</body>
</html>
```



- Oefening
  - Voeg een nieuwe html pagina toe
    - Test.html (inhoudelijk: Dit is een testpagina!)
  - Voeg een hyperlink toe in default.html die verwijst naar test.html



- Activeer middleware voor Razor pages
  - Add Razor page
    - Index.cshtml

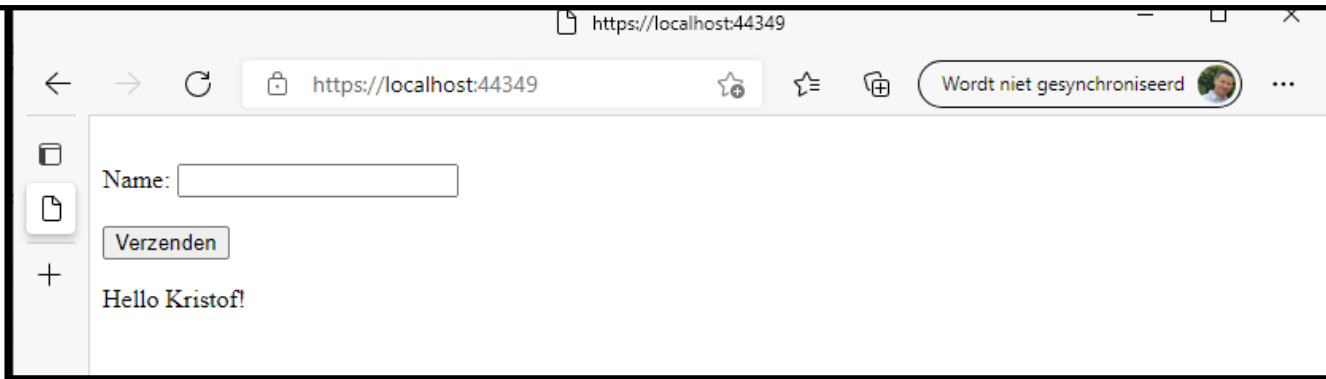




- Project instellingen
  - Add Folder
    - Pages
      - Add razor page
        - index.cshtml
- Activeer middleware voor HTML content
  - Startup.cs
    - ConfigureServices

```
services.AddControllers();
services.AddRazorPages();
```
    - Configure

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapRazorPages();
});
```



```

@page

@{
    var name = string.Empty;
    if (Request.HasFormContentType)
    {
        name = Request.Form["name"];
    }
}

<div style="margin-top:30px;">
    <form method="post">
        <div>Name: <input name="name" /></div>
        <br />
        <div><input type="submit" /></div>
        @Html.AntiForgeryToken()
    </form>
</div>
<div>
    @if (!string.IsNullOrEmpty(name))
    {
        <p>Hello @name!</p>
    }
</div>

```

### Oefening Razor Pages

- Voeg een 2<sup>de</sup> razor page toe in je webapplicatie
  - Jouwvoornaam.cshtml (Kristof.cshtml)
  - Maak een hyperlink van je index.cshtml naar de nieuwe pagina

## Oefening Razor Pages

- Voeg een 2<sup>de</sup> razor page toe in je webapplicatie
  - Jouwvoornaam.cshtml (Kristof.cshtml)
  - Maak een hyperlink van je index.cshtml naar de nieuwe pagina

Pages – Add file

- \_ViewImports.cshtml

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers