https://www.halvorsen.blog

ASP.NET Core

Hans-Petter Halvorsen

Table of Contents

- Introduction to ASP.NET Core
- ASP.NET Core in Visual Studio
- ASP.NET Core - Hello World Application
- ASP.NET Core - Fundamentals
- ASP.NET Core Razor

ASP.NET Core

Introduction

ASP.NET

Different Programming models:

- ASP.NET Web Forms
- ASP.NET MVC (Model-View-Controller)
- **ASP.NET Razor Pages**
  - Razor Single Page Model (comparable to PHP but using C# syntax instead)
  - Razor with Page Model (Code and Layout are separated in different Files)
  - Razor with MVC

ASP.NET Razor

Different Types

- Razor Page (Single Page Model)
  - Test1.cshtml
- **Razor Page (with Page Model)**
  - **Test2.cshtml + Test2.cshtml.cs**
- Razor with MVC

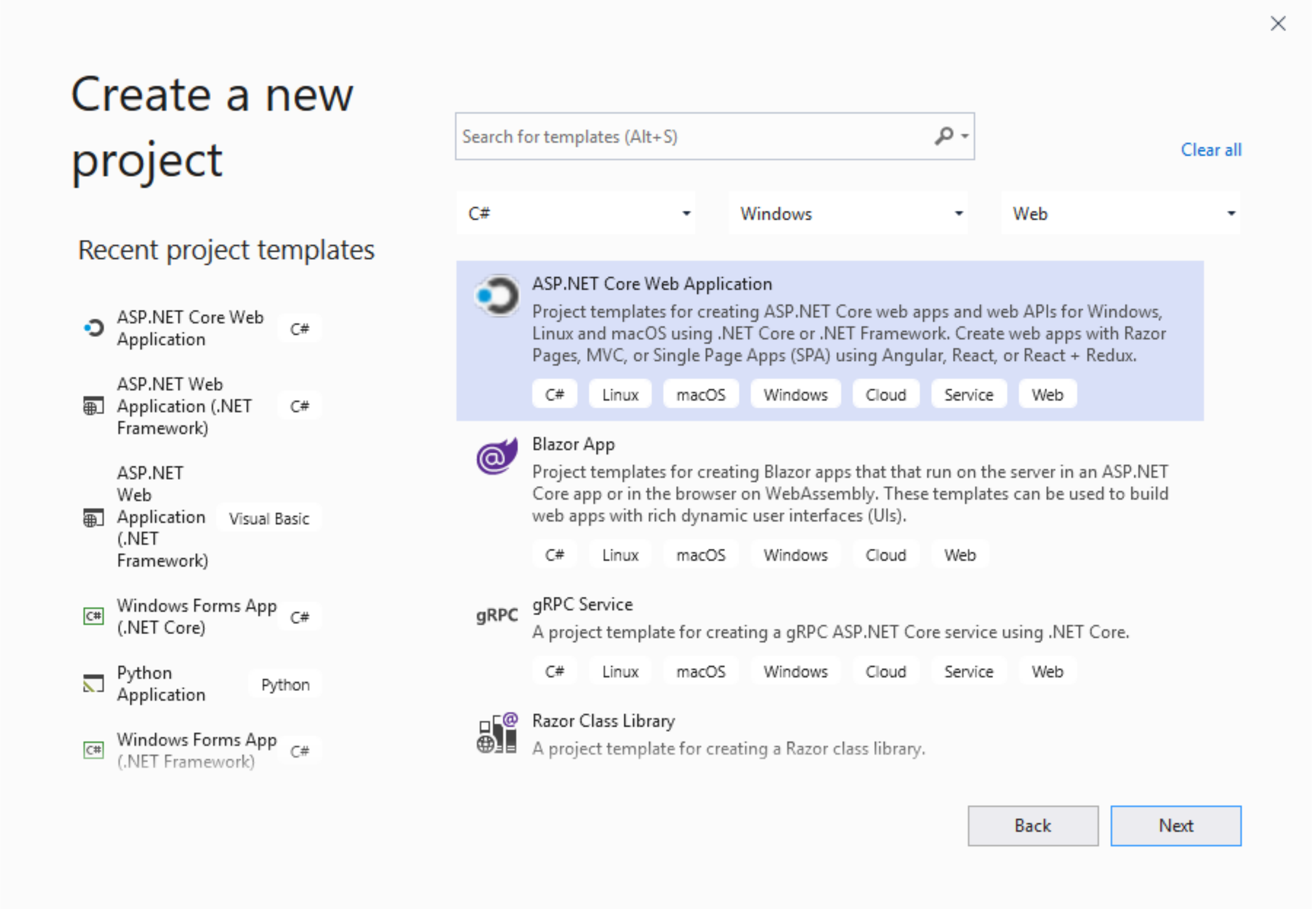
Razor Pages

Each Razor page is a pair of files:

- A ***.cshtml*** file that contains HTML markup with C# code using Razor syntax.
- A ***.cshtml.cs*** file that contains C# code that handles page events.

ASP.NET Core

Visual Studio



Create a new project

Search for templates (Alt+S)

Clear all

C# | Windows | Web

Recent project templates

- ASP.NET Core Web Application — C#
- ASP.NET Web Application (.NET Framework) — C#
- ASP.NET Web Application (.NET Framework) — Visual Basic
- Windows Forms App (.NET Core) — C#
- Python Application — Python
- Windows Forms App (.NET Framework) — C#

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 that run on the server in an ASP.NET Core app or in the browser on WebAssembly. These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C# · Linux · macOS · Windows · Cloud · Web

gRPC Service

A project template for creating a gRPC ASP.NET Core service using .NET Core.

C# · Linux · macOS · Windows · Cloud · Service · Web

Razor Class Library

A project template for creating a Razor class library.

Folders an Files

- appSettings.json
  - Contains configuration data, such as connection strings
- Program.cs
  - Contains the entry point for the program
- Startup.cs
  - Contains code that configures app behavior.
- **wwwroot** folder
  - Contains static files, such as HTML files, JavaScript files, and CSS files
- **Pages** folder
  - You may put your ASP.NET web pages here

Supporting Files

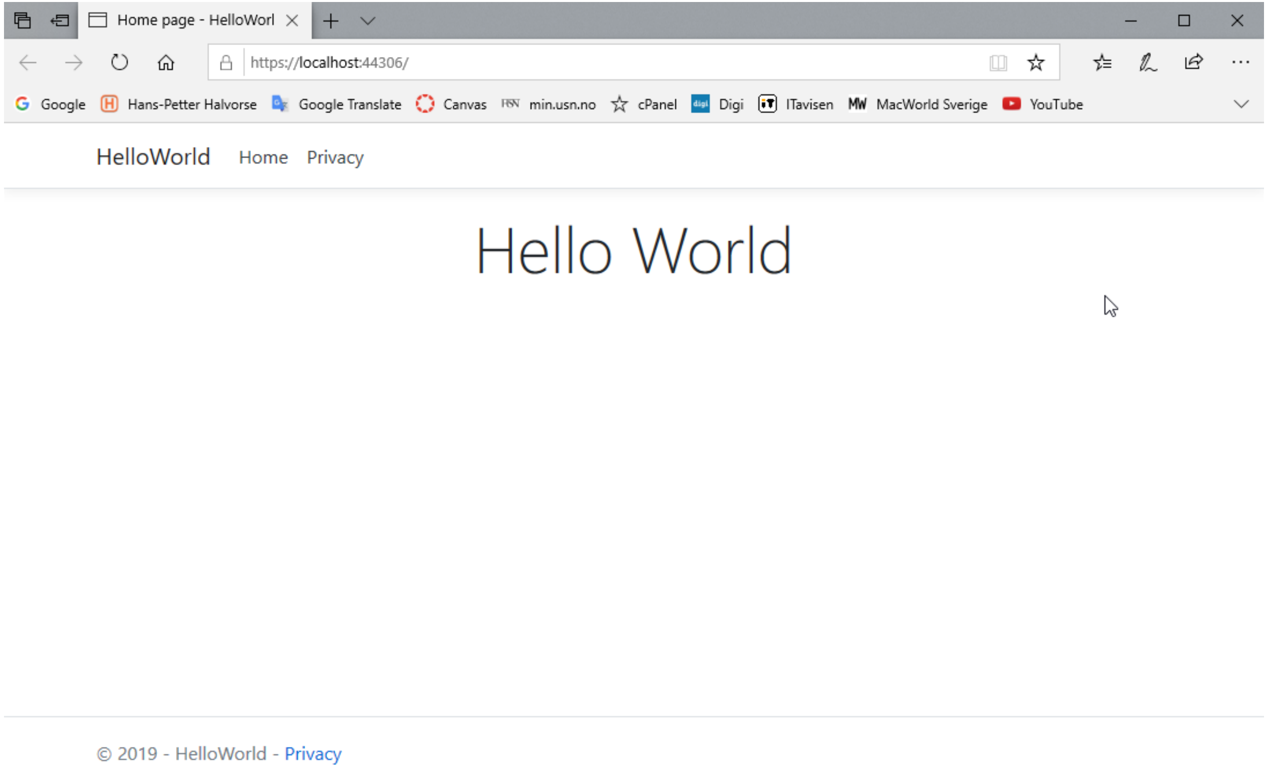
Supporting files have names that begin with an underscore (_).

- ***_Layout.cshtml*** file configures UI elements common to all pages. You can use this file to set up the navigation menu at the top of the page

ASP.NET Core

Hello World Application

Hello World Application



Home page - HelloWorl

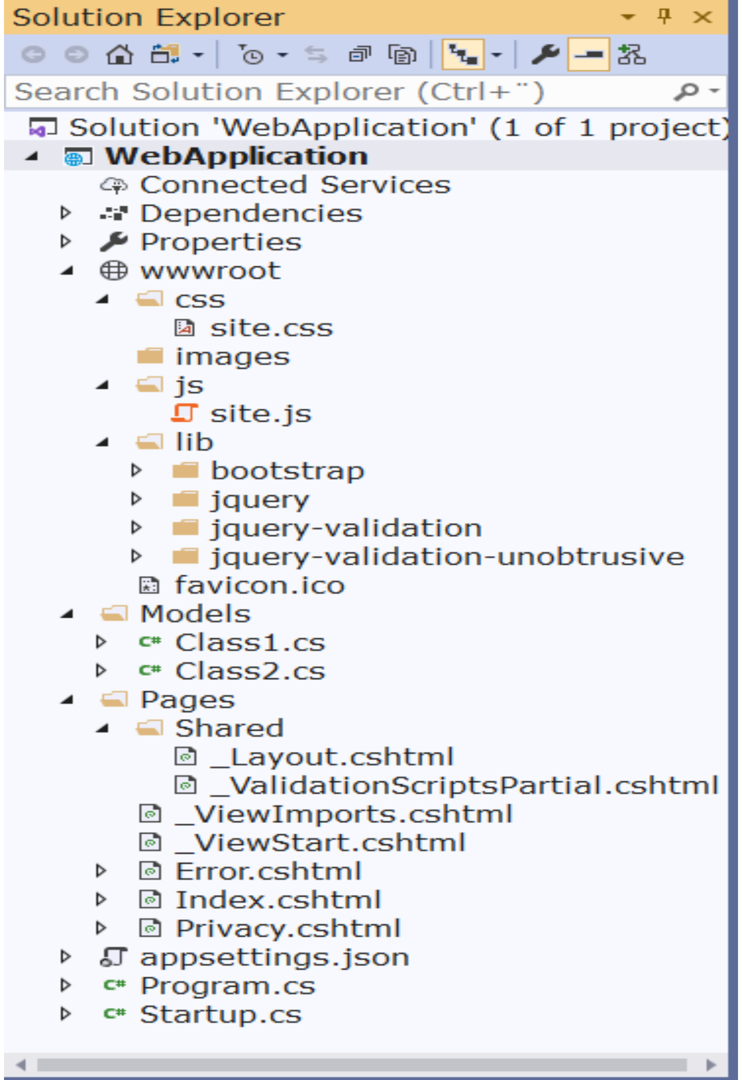
https://localhost:44306/

Hello World

© 2019 - HelloWorld - Privacy

ASP.NET Core

Fundamentals



Solution Explorer

Search Solution Explorer (Ctrl+¨)

```
Solution 'WebApplication' (1 of 1 project)
  WebApplication
    Connected Services
    Dependencies
    Properties
    wwwroot
      css
        site.css
      images
      js
        site.js
      lib
        bootstrap
        jquery
        jquery-validation
        jquery-validation-unobtrusive
      favicon.ico
    Models
      Class1.cs
      Class2.cs
    Pages
      Shared
        _Layout.cshtml
        _ValidationScriptsPartial.cshtml
      _ViewImports.cshtml
      _ViewStart.cshtml
      Error.cshtml
      Index.cshtml
      Privacy.cshtml
    appsettings.json
    Program.cs
    Startup.cs
```

- appSettings.json
  - Contains configuration data, such as connection strings
- Program.cs
  - Contains the entry point for the program
- Startup.cs
  - Contains code that configures app behavior.
- **wwwroot** folder
  - Contains static files, such as HTML files, JavaScript files, and CSS files
- **Pages** folder
  - You may put your ASP.NET web pages here
- **Models** folder
  - Here you create your Classe

Folders an Files

- appSettings.json
  - Contains configuration data, such as connection strings
- Program.cs
  - Contains the entry point for the program
- Startup.cs
  - Contains code that configures app behavior.
- **wwwroot** folder
  - Contains static files, such as HTML files, JavaScript files, and CSS files
- **Pages** folder
  - You may put your ASP.NET web pages here
- **Models** folder
  - Here you create your Classes

Supporting Files

Supporting files have names that begin with an underscore (_).

- ***_Layout.cshtml*** file configures UI elements common to all pages. You can use this file to set up the navigation menu at the top of the page

Pages Folder

- You should put your ASP.NET web pages here

Models Folder

- Here you should create your Classes

appSettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",

  "ConnectionStrings": {
    "ConnectionString": "DATA SOURCE=xxx;UID=xxx;PWD=xxx;DATABASE=xxx"
  }

 }
```

ASP.NET Core

Razor

Razor

- Razor is a markup syntax for embedding server-based code into webpages.
- The Razor syntax consists of Razor markup, C#, and HTML. Files containing Razor generally have a **.cshtml** file extension.
- The default Razor language is HTML. Rendering HTML from Razor markup is no different than rendering HTML from an HTML file. HTML markup in .cshtml Razor files is rendered by the server unchanged.

ASP.NET Razor

Different Types

- Razor Page (Single Page Model)
  - Test1.cshtml
- **Razor Page (with Page Model)**
  - **Test2.cshtml + Test2.cshtml.cs**
- Razor with MVC

Razor Pages

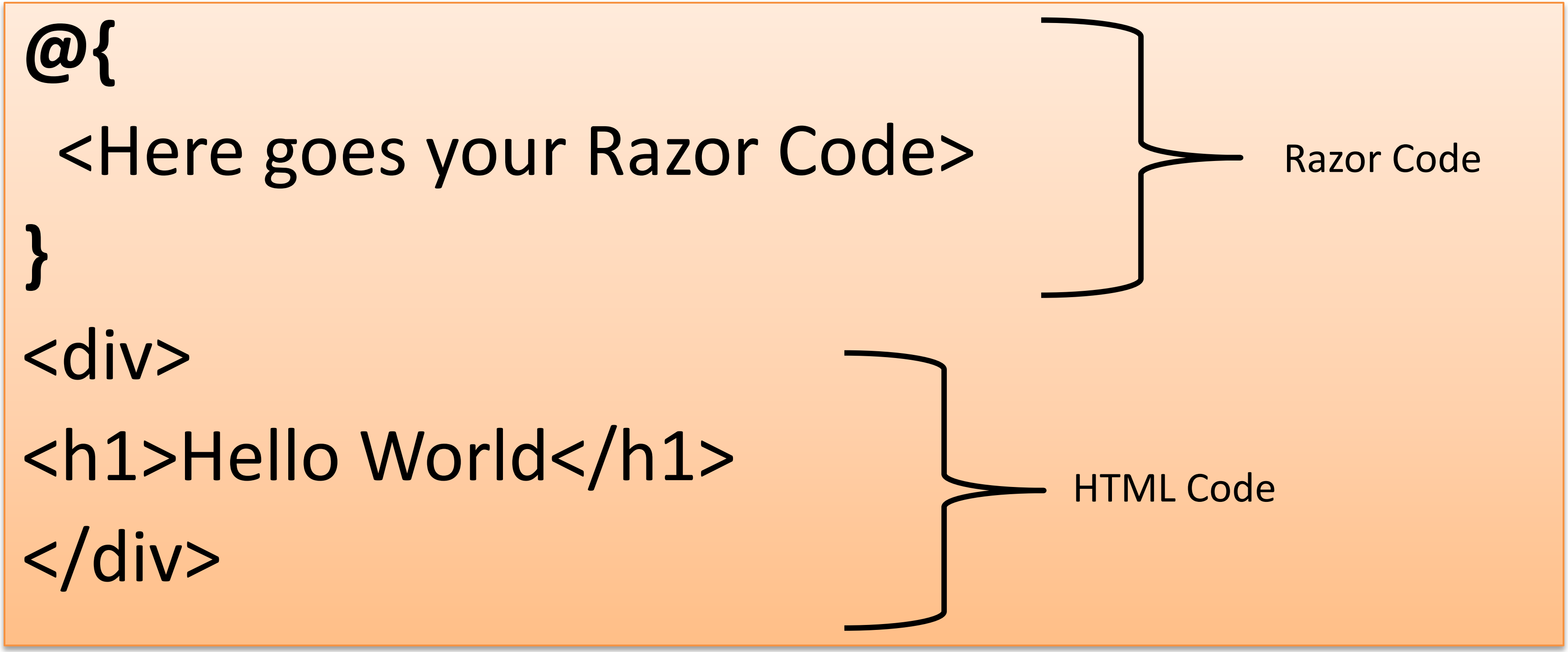
Each Razor page is a pair of files:

- A ***.cshtml*** file that contains HTML markup with C# code using Razor syntax.
- A ***.cshtml.cs*** file that contains C# code that handles page events.

Razor Syntax

- Razor supports C# and uses the **@** symbol to transition from HTML to C#.
- Razor evaluates C# expressions and renders them in the HTML output.
- All code blocks must appear within **@{ … }** brackets.

Razor Syntax



```
@{
 <Here goes your Razor Code>
}
<div>
<h1>Hello World</h1>
</div>
```

ASP.NET Core

Database Communication

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog