

<https://www.halvorsen.blog>

ASP.NET Core Basics



ASP.NET Core

Form Data - Retrieving User Data from Client to Server

Hans-Petter Halvorsen

- ASP.NET Core
- Practical Example
 - Retrieving Data from Client to Server
 - Form

This Tutorial are for rookies when it comes to Web Programming and ASP.NET Core

An **HTML form** is used to collect user input. The user input is most often sent to a server for processing.

FormDataApp [Home](#) [Privacy](#)

Welcome

Please enter your Name:

Name:

Your Name is: Hans-Petter Halvorsen

© 2021 - FormDataApp - [Privacy](#)



ASP.NET Core

Hans-Petter Halvorsen

[Table of Contents](#)

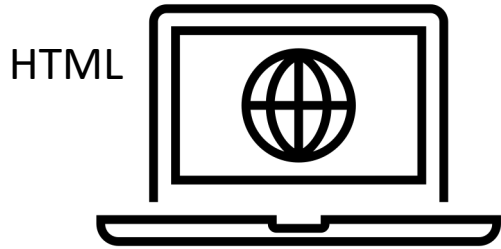
ASP.NET Core

- ASP.NET Core is an open-source web framework, created by Microsoft, for building web applications and web services
- You create ASP.NET Core Web Applications using Visual Studio and the C# Programming language
- ASP.NET Resources:

<https://www.halvorsen.blog/documents/programming/web/aspnet>

Web Fundamentals

Client-side

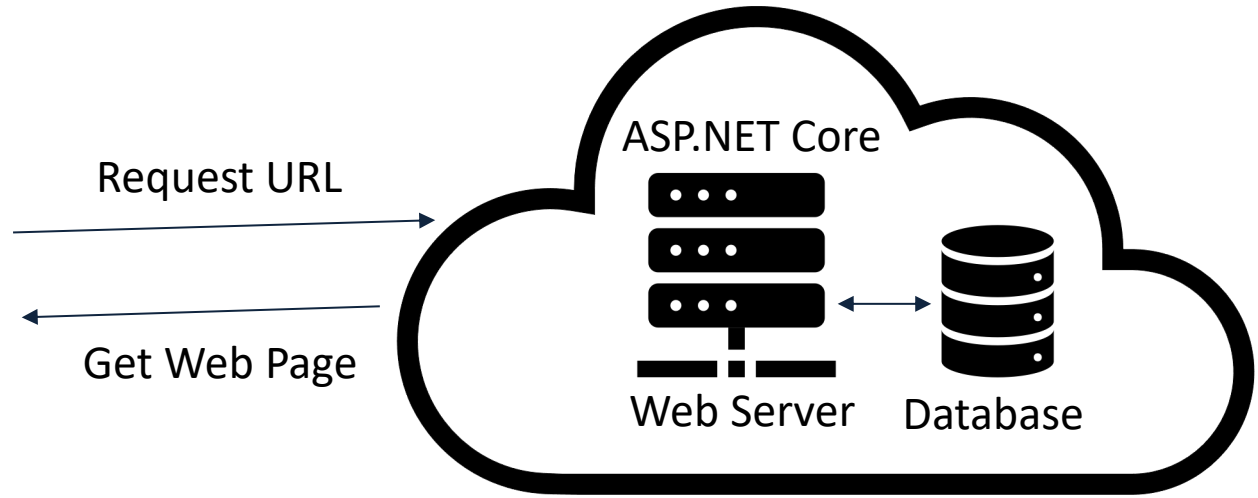


HTML

Local Computer
with Web Browser

A pure HTML File is sent to
your local Web Browser

Server-side



The Cloud/Internet

Server-side Code (C#) is executed on the Server

Web Fundamentals

HTML

CSS

JavaScript

Client-side

Server-side
Framework

HTML is the standard markup language for creating Web pages

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

JavaScript is the programming language of the Web. JavaScript is probably the most used programming language in the world.

ASP.NET Core (C#), PHP or similar



Code Example

ASP.NET Core Example

FormDataApp [Home](#) [Privacy](#)

Welcome

Please enter your Name:

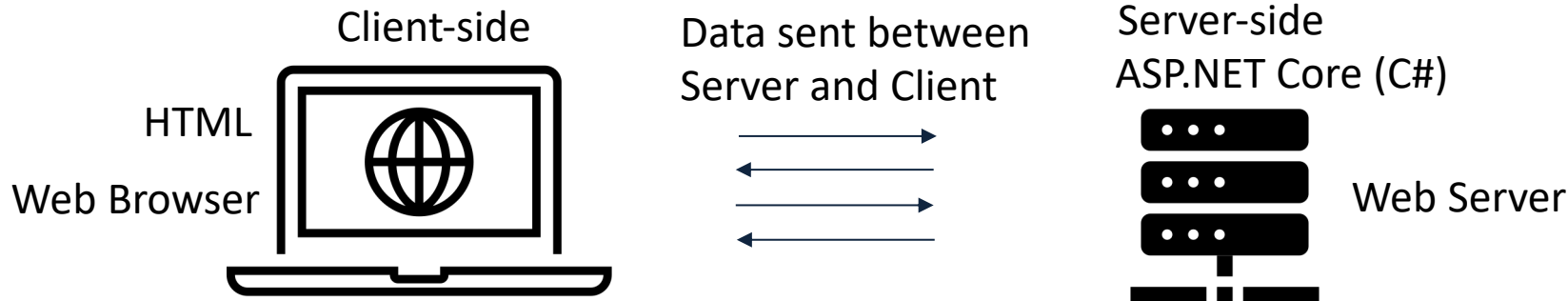
Name:

OK

Your Name is: Hans-Petter Halvorsen

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

ASP.NET Core Example



FormDataApp Home Privacy

Welcome

Please enter your Name:

Name:

OK

Your Name is:

© 2021 - FormDataApp - Privacy

FormDataApp Home Privacy

Welcome

Please enter your Name:

Name:

OK

Your Name is: Hans-Petter Halvorsen

© 2021 - FormDataApp - Privacy

ASP.NET Core Template

Configure your new project

ASP.NET Core Web Application

Project name

FormDataApp

Location

C:\Users\hansha\OneDrive\Programming\Demo A

Solution name ⓘ

FormDataApp

☐ Place solution and project in the same directory

Create a new ASP.NET Core web ap...

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



ASP.NET Core Web 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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

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

Linux

☐ Enable Razor runtime compilation

Author: Microsoft

Source: Templates 5.0.4

Back

Create

Back

Create

ASP.NET Core Template

The screenshot displays the Visual Studio IDE with a new ASP.NET Core web application project named 'FormDataApp'. The main editor shows the 'Index.cshtml' file, which contains the following code:

```
1 @page
2 @model IndexModel
3 @{
4     ViewData["Title"] = "Welcome";
5 }
6
7 <div>
8
9     <h1>Welcome</h1>
10
11     <p>Please enter your Name:</p>
12
13     <form name="NameForm" id="NameForm" method="post">
14
15         Name:
16         <br />
17
18         <input name="FullName" type="text" class="form-control input-lg autofocus required" />
19         <br />
20
21         <input id="OKButton" type="submit" value="OK" class="btn btn-info" />
22
23     </form>
24     <br />
25
26     <p>
27         Your Name is: @Model.FullName
28     </p>
29
30 </div>
```

The left sidebar shows the 'Toolbox' with HTML and General controls. A message states: 'There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.' The right sidebar shows the 'Solution Explorer' with the project structure:

- Solution 'FormDataApp' (1)
- FormDataApp
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Pages
 - Shared
 - _ViewImports.cshtml
 - _ViewStart.cshtml
 - Error.cshtml
 - Index.cshtml
 - Index.cshtml.cs
 - Privacy.cshtml
 - appsettings.json
 - Program.cs
 - Startup.cs

The bottom status bar shows '110 %', 'No issues found', and 'Ln: 11 Ch: 5 SPC CRLF'. The 'Error List' is empty, showing '0 Errors', '0 Warnings', and '0 Messages'. The 'Output' window is also empty.

Index Page

FormDataApp [Home](#) [Privacy](#)

Welcome

Please enter your Name:

Name:

OK

Your Name is:

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

```
<div>
```

```
<h1>Welcome</h1>
```

```
<p>Please enter your Name:</p>
```

```
<form name="NameForm" id="NameForm" method="post">
```

```
    Name:
    <br />
```

```
    <input name="FullName" type="text" class="form-control input-lg" autofocus required />
    <br />
```

```
    <input id="OKButton" type="submit" value="OK" class="btn btn-info" />
```

```
</form>
<br />
```

```
<p>Your Name is: @Model.fullName</p>
```

```
</div>
```

FormDataApp Home Privacy

Welcome

Please enter your Name:

Name:

Your Name is:

© 2021 - FormDataApp - [Privacy](#)

Index.cshtml

`<input name="FullName" type="text" />`

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

```
using Microsoft.AspNetCore.Mvc.RazorPages;
```

```
namespace FormDataApp.Pages
```

```
{
```

```
    public class IndexModel : PageModel  
    {
```

```
        public string fullName;
```

```
        public void OnGet()  
        {  
  
        }
```

```
        public void OnPost()  
        {
```

```
            fullName = Request.Form["FullName"];
```

```
        }
```

```
    }
```

```
}
```

Make sure to set the variables to **public** to make them available in the web page

The **OnPost()** method is automatically called when you push the Button

To receive Data from the Web Page we use **Request.Form["xxx"]**

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: <https://www.halvorsen.blog>

