About the Tutorial

ASP.NET Core is the new web framework from Microsoft. ASP.NET Core is the framework you want to use for web development with .NET. At the end this tutorial, you will have everything you need to start using ASP.NET Core and write an application that can create, edit, and view data from a database.

Audience

This tutorial is designed for software programmers who would like to learn the basics of ASP.NET Core from scratch.

Prerequisites

You should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages is a plus.

Disclaimer & Copyright

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.
# Table of Contents

- About the Tutorial .................................................................................................................. i
- Audience ................................................................................................................................... i
- Prerequisites .............................................................................................................................. i
- Disclaimer & Copyright ............................................................................................................ i
- Table of Contents ....................................................................................................................... ii

1. ASP.NET — OVERVIEW ........................................................................................................ 1
   - A Brief History of ASP.NET ................................................................................................... 1
   - What is ASP.NET Core .......................................................................................................... 1
   - Advantages of ASP.NET Core .............................................................................................. 2

2. ASP.NET CORE — ENVIRONMENT SETUP ........................................................................... 3
   - Installation of Microsoft Visual Studio 2015 ....................................................................... 3

3. ASP.NET CORE — NEW PROJECT ......................................................................................... 9

4. ASP.NET CORE — PROJECT LAYOUT .................................................................................. 13

5. ASP.NET CORE — PROJECT.JSON ..................................................................................... 20

6. ASP.NET CORE — CONFIGURATION .................................................................................... 25

7. ASP.NET CORE — MIDDLEWARE .......................................................................................... 30
   - How to Add another Middleware ......................................................................................... 33

8. ASP.NET CORE — EXCEPTIONS ........................................................................................... 37

9. ASP.NET CORE — STATIC FILES ....................................................................................... 41

10. ASP.NET CORE — SETUP MVC .......................................................................................... 49
11. **ASP.NET Core — MVC Design Pattern** ................................................................. 56
   Idea behind MVC ................................................................................................. 56

12. **ASP.NET Core — Routing** ........................................................................... 59
13. **ASP.NET Core — Attribute Routes** ............................................................... 68
14. **ASP.NET Core — Action Results** ................................................................. 73
15. **ASP.NET Core — Views** ............................................................................... 79
16. **ASP.NET Core — Setup Entity Framework** .................................................. 85
17. **ASP.NET Core — DbContext** ......................................................................... 89
18. **ASP.NET Core — Razor Layout Views** ......................................................... 104
19. **ASP.NET Core — Razor View Start** ............................................................... 109
20. **ASP.NET Core — Razor View Import** ............................................................ 112
21. **ASP.NET Core — Razor Tag Helpers** ............................................................ 117
22. **ASP.NET Core — Razor Edit Form** ............................................................... 125
23. **ASP.NET Core — Identity Overview** ............................................................. 135
24. **ASP.NET Core — Authorize Attribute** .......................................................... 138
25. **ASP.NET Core — Identity Configuration** ..................................................... 149
26. **ASP.NET Core — Identity Migrations** .......................................................... 160
27. **ASP.NET Core — User Registration** ............................................................... 165
28. **ASP.NET Core — Create A User** ................................................................. 173
29. **ASP.NET Core — Log In And Log Out** ......................................................... 178
ASP.NET Core is the new web framework from Microsoft. It has been redesigned from the ground up to be fast, flexible, modern, and work across different platforms. Moving forward, ASP.NET Core is the framework that can be used for web development with .NET. If you have any experience with MVC or Web API over the last few years, you will notice some familiar features. At the end this tutorial, you will have everything you need to start using ASP.NET Core and write an application that can create, edit, and view data from a database.

A Brief History of ASP.NET

ASP.NET has been used from many years to develop web applications. Since then, the framework went through a steady evolutionary change and finally led us to its most recent descendant ASP.NET Core 1.0.

- ASP.NET Core 1.0 is not a continuation of ASP.NET 4.6.
- It is a whole new framework, a side-by-side project which happily lives alongside everything else we know.
- It is an actual re-write of the current ASP.NET 4.6 framework, but much smaller and a lot more modular.
- Some people think that many things remain the same, but this is not entirely true. ASP.NET Core 1.0 is a big fundamental change to the ASP.NET landscape.

What is ASP.NET Core

ASP.NET Core is an open source and cloud-optimized web framework for developing modern web applications that can be developed and run on Windows, Linux and the Mac. It includes the MVC framework, which now combines the features of MVC and Web API into a single web programming framework.

- ASP.NET Core apps can run on .NET Core or on the full .NET Framework.
- It was architected to provide an optimized development framework for apps that are deployed to the cloud or run on-premises.
- It consists of modular components with minimal overhead, so you retain flexibility while constructing your solutions.
• You can develop and run your ASP.NET Core apps cross-platform on Windows, Mac and Linux.

Advantages of ASP.NET Core

ASP.NET Core comes with the following advantages:

• ASP.NET Core has a number of architectural changes that result in a much leaner and modular framework.
• ASP.NET Core is no longer based on System.Web.dll. It is based on a set of granular and well factored NuGet packages.
• This allows you to optimize your app to include just the NuGet packages you need.
• The benefits of a smaller app surface area include tighter security, reduced servicing, improved performance, and decreased costs.

With ASP.NET Core, you can get the following improvements:

• Build and run cross-platform ASP.NET apps on Windows, Mac and Linux
• Built on .NET Core, which supports true side-by-side app versioning
• New tooling that simplifies modern web development
• Single aligned web stack for Web UI and Web APIs
• Cloud-ready environment-based configuration
• Built-in support for dependency injection
• Tag Helpers which makes Razor markup more natural with HTML
• Ability to host on IIS or self-host in your own process
ASP.NET Core is a significant redesign of ASP.NET. This topic introduces the new concepts in ASP.NET Core and explains how they help you develop modern web apps.

To use ASP.NET Core in your application, the following must be installed in your system:

- Microsoft Visual Studio 2015
- Microsoft .NET Core 1.0.0 - VS 2015 Tooling Preview 2

Microsoft provides a free version of Visual Studio which also contains the SQL Server and it can be downloaded from [https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx](https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx) and Microsoft .NET Core 1.0.0 - VS 2015 Tooling Preview 2 can be downloaded from [https://go.microsoft.com/fwlink/?LinkId=817245](https://go.microsoft.com/fwlink/?LinkId=817245).

### Installation of Microsoft Visual Studio 2015

Let us now understand the steps involved in the installation of

**Step 1:** Once downloading is completed, run the installer. The following dialog box will be displayed.
Choose your installation location
C:\Program Files (x86)\Microsoft Visual Studio 14.0
Setup requires up to 6 GB across all drives.

Choose the type of installation
- Typical
  Includes C#/.NET Web and Desktop features
- Custom
  Allows you to customize features for your installation

You can add or remove additional features at any time after setup via Programs and Features in the Control Panel.

By clicking the “Install” button, I acknowledge that I accept the License Terms and Privacy Statement.
Step 2: Click the Install button as in the above screenshot. The installation process will then start.
**Step 3:** Once the installation process is completed successfully, you will see the following dialog box.

![Visual Studio Setup Completed dialog box](image)

**Step 4:** Close this dialog and restart your computer if required.
Step 5: Open the Visual studio from the Start menu. This will open the following dialog box and it will take some time for the first time (only for preparation).
**Step 6:** You will now see the main window of the Visual studio.

![Visual Studio Main Window](image)

**Step 7:** Once the Visual Studio is installed, close the Visual Studio and launch the Microsoft .NET Core 1.0.0 - VS 2015 Tooling Preview 2
**Step 8:** Check the checkbox and click Install.

**Step 9:** Once the installation is completed, you will see the following message.
Step 10: You are now ready to start your application using ASP.NET Core.
In this chapter, we will discuss how to create a new project in Visual Studio.

Once you have installed the Visual Studio 2015 tooling, you can start building a new ASP.NET Core Application from the File > New Project menu option.

On the New Project dialog box, you will see the following three different templates for Web projects:

- **ASP.NET Web Application**: The simple ASP.NET application templates
- **ASP.NET Core Web Application (.NET Core)**: This will start you with a cross-platform compatible project that runs on the .NET Core framework
- **ASP.NET Core Web Application (.NET Framework)**: This starts a new project that runs on the standard .NET Framework on Windows.
In the left pane, select **Templates > Visual C# > Web** and in the middle pane select the ASP.NET Core Web Application (.NET Core) template. Let us call this application **FirstAppDemo** and also specify the Location for your ASP.NET Core project and then Click OK.

In the above dialog box, you can select a specific template for the ASP.NET application from the available ASP.NET Core Templates.

ASP.NET Core templates currently contain three different templates. Of these, the **Web Application template** will help you lay out a lot of files on your file system. This also allows you to use ASP.NET MVC right away.

Here, we will start with an empty template. This would help us build it from scratch. Let us select the Empty template, turn off the Host in the cloud and click OK.
Visual Studio will now launch the project in some time. In the Solution Explorer window, you will see all the files that are in this project.

Let us run this application, you can do that by pressing Ctrl+F5 or by going to the Debug menu. After going to the Debug menu, select **Start Without Debugging**.
This application can display only Hello World! This runs on **localhost:57741**. In the window system tray, you can also see that IIS Express is running.
And the name of the site is **FirstAppDemo**. If you have been programming with ASP.NET with the previous versions of the framework, the way you would interact with the Visual Studio and the way Visual Studio uses IIS Express to host your application, all these aspects will be familiar.
In this chapter, we will discuss how ASP.NET core project appears on the file system and how the different files and directories all work together.

Let us open the **FirstAppDemo** project created in the previous chapter.
In the Solution Explorer window, right-click on the Solution node and select the Open Folder in File Explorer.

You will see now the root directory with two files in it: FirstAppDemo.sln and global.json.
FirstAppDemo.sln is a solution file. Visual Studio has used this extension for years by default, and you can double-click on the file if you want to open the app in Studio and work on it.

There is also a global.json file. Let us open this file in Visual Studio.
In the file, the project's setting is significant. This project setting tells ASP.NET where to look for your source code and what folders contain your projects.

There are two possible folders “src” for source and a “test” folder. Unless your projects and source code are in one of these two folders, the code won't be available to build. You can change these settings if you want.

The Windows Explorer has the “src” folder on disk. You don’t have a test folder. In the test folder, you could place your unit testing projects. Let us double-click on the “src” folder.
You can see the FirstAppDemo project, and the web application. Now, double-click on the folder.
These are the source code files for the application and you can also see this folder structure in the Solution Explorer window. This is because in the present version of ASP.NET Core, the file system determines what is in your project.
End of ebook preview
If you liked what you saw...
Buy it from our store @ https://store.tutorialspoint.com