ASP.NET 6 New Features

https://www.accelebrate.com
(877) 849-1850 ❖ info@accelebrate.com

Customized Technical Training

Accelebrate
ACCELERATED LEARNING, CELEBRATED RESULTS®

On-Site, Customized Private Training

Don't settle for a one-size-fits-all class! Let Accelebrate tailor a private class to your group's goals and experience. Classes can be delivered at your site or online (or a combination of both) worldwide. Visit us at **https://www.accelebrate.com** and contact us at **sales@accelebrate.com** for details.

Public Online Training

Need to train just 1-3 people? Attend one of our regularly scheduled, live, instructor-led public online classes. For course dates, times, outlines, pricing, and registration, visit **https://www.accelebrate.com/public-training-schedule.**

Newsletter

Want to find out about our latest class offerings? Subscribe to our newsletter **https://www.accelebrate.com/newsletter**.

Blog

Get insights and tutorials from our instructors and staff! Visit our blog, **https://www.accelebrate.com/blog** and join the discussion threads and get feedback from our instructors!

Learning Resources

Get access to learning guides, tutorials, and past issues of our newsletter at the Accelebrate library, **https://www.accelebrate.com/library**.

Call us for a training quote!

877 849 1850

aws PARTNER

Microsoft Learning Partner

Accelebrate, Inc. was founded in 2002 with the goal of delivering **private training** that rapidly achieves participants' goals. Each year, our experienced instructors deliver hundreds of classes online and at client sites all over the US, Canada, and abroad. We pride ourselves on our **instructors' real-world experience** and ability to **adapt the training** to your team and their objectives. We offer a wide range of topics, including:

- AWS, Azure, and Cloud Computing
- Angular, React, and Vue
- JavaScript
- Data Science using R, Python, & Julia
- Excel Power Query
- Power BI & Tableau
- .NET & VBA programming
- SharePoint & Microsoft 365
- DevOps & CI/CD
- iOS & Android Development
- PostgreSQL, Oracle, and SQL Server
- Java, Spring, and Groovy
- Agile, DEI, & IT Leadership
- Web/Application Server Admin
- HTML5 & Mobile Web Development
- Adobe & Articulate
- Docker, Kubernetes, Ansible, & Git
- Software Design and Testing
- AND MORE *(see back)*

"Our organization and members love Accelebrate! Every training opportunity has been well-received, well-presented, and easy to set up. Not only is Accelebrate a great training corporation, but the customer service is unmatched."

— Emily, MnCCC

Like us on Facebook + Follow us on Twitter + Watch us on YouTube

Visit our website for a complete list of courses!

Adobe & Articulate
Adobe Captivate
Adobe Presenter
Articulate Storyline / Studio
Camtasia
RoboHelp

AWS, Azure, & Cloud
AWS
Azure
Google Cloud
OpenStack
Terraform
VMware

Big Data
Alteryx
Apache Spark
Teradata
Snowflake SQL

Data Science and RPA
Apache Airflow
Blue Prism
Data Literacy
Django
Julia
Machine Learning
MATLAB
Minitab
Python
R Programming
SPSS
UiPath

Data Visualization
BusinessObjects
Crystal Reports
Excel Power Query
Power BI
PivotTable and PowerPivot
Qlik
Tableau

Database
MongoDB
NoSQL Databases
Oracle
Oracle APEX
PostgreSQL
SQL Server
Vertica Architecture & SQL

DevOps, CI/CD & Agile
Agile
Ansible
Apache Maven
Chef
DEI
Docker
Git
IT Leadership
ITIL
Jenkins
Jira & Confluence
Kubernetes
Linux
Microservices
OpenShift
Six Sigma
Software Design

Java
Groovy and Grails
Java & Web App Security
JavaFX
JBoss
Scala
Selenium & Cucumber
Spring Boot
Spring Framework

JS, HTML5, & Mobile
Angular
CSS
D3.js
Flutter
HTML5
iOS/Swift Development
JavaScript
Node.js & Express
React & Redux
Svelte
Swift
Symfony
Xamarin
Vue

Microsoft & .NET
.NET Core
ASP.NET
Azure DevOps
Blazor
C#
Design Patterns
Entity Framework Core
IIS
Microsoft Dynamics CRM
Microsoft 365
Microsoft Power Platform
Microsoft Project
Microsoft SQL Server
Microsoft System Center
Microsoft Windows Server
PowerPivot
PowerShell
VBA
Visual C++/CLI
Visual Studio
Web API

Security
.NET Web App Security
C and C++ Secure Coding
C# & Web App Security
Linux Security Admin
Python Security
Secure Coding for Web Dev
Spring Security

SharePoint
Power Automate & Flow
SharePoint Administrator
SharePoint Developer
SharePoint End User
SharePoint Online
SharePoint Site Owner

SQL Server
Azure SQL Data Warehouse
Business Intelligence
Performance Tuning
SQL Server Administration
SQL Server Development
SSAS, SSIS, SSRS
Transact-SQL

Teleconferencing Tools
Adobe Connect
GoToMeeting
Microsoft Teams
WebEx
Zoom

Web/Application Server
Apache httpd
Apache Tomcat
IIS
JBoss
Oracle WebLogic

Other
C++
Go Programming
Mulesoft
Project Management
Ruby on Rails
Rust
Salesforce
Sitefinity
UX

*Visit **www.accelebrate.com/newsletter** to sign up and receive our **newsletters** with information about **new courses**, **free webinars**, **tutorials**, and **blog articles**.*

Call us for a training quote! 877 849 1850 (US/Canada) or +1 678 648 3133

What's New in ASP.NET Core 6

Agenda

- Introduction
- What's New in C# 10
- Performance Improvements
- ASP.NET Core Architecture
- Migrating an Existing Application to .NET 6
- Conclusion and Q&A

What's New in ASP.NET Core 6

Introduction to .NET

- Evolution of the .NET Platform
- .NET SDK and Runtimes
- Visual Studio

2002 .NET 1.0, C# 1.0, Visual Studio .NET

2003 .NET 1.1, Visual Studio 2003

2004 Mono 1.0

2005 .NET 2.0, C# 2.0, Visual Studio 2005
Generics, Nullable Value Types

2006 .NET 3.0, Mono 1.2
WPF, WCF, WF

2007 .NET 3.5, C# 3.0, Visual Studio 2008
LINQ, Anonymous Types, Lambda Expressions, Extension Methods, Implicit Typing

2008 Entity Framework 1.0

2009 ASP.NET MVC 1.0

2010 .NET 4.0, C# 4.0, ASP.NET MVC 2, Visual Studio 2010
Named / Optional Arguments, Dynamic Binding

2012 .NET 4.5, C# 5.0, Mono 3.0, ASP.NET MVC 4, Visual Studio 2012
Asynchronous Members (async / await)

2013 .NET 4.5.1, ASP.NET MVC 5, Visual Studio 2013
SignalR 1.0

2015 .NET 4.6, C# 6.0, Mono 4.0, Visual Studio 2015, Visual Studio Code 1.0
Expression Bodied Members, Null Propagator, String Interpolation

2016 Xamarin Acquisition, .NET Core 1.0, .NET Standard 1.0
Entity Framework Core 1.0

2017 .NET 4.7, .NET Core 2.0, C# 7.0, Visual Studio 2017
ASP.NET Razor Pages, Out Variables, Tuples, Ref Locals and Returns

2018 GitHub Acquisition, .NET Standard 2.0
Blazor Server

2019 .NET 4.8, .NET Core 3.0, C# 8.0, Visual Studio 2019
gRPC, Default Interface Methods, Using Declarations, Nullable Reference Types

2020 .NET 5, C# 9.0
Blazor WebAssembly, Records, Init Only Setters, Top-Level Statements

2021 .NET 6, C# 10.0, Visual Studio 2022
.NET MAUI

Introduction to .NET

Evolution of the .NET Platform

- The version of .NET Core after 3.1 became the "main line" for .NET and was labeled .NET 5.0
- In .NET 5 and .NET 6, the ASP.NET framework still includes the name "Core" to avoid confusion with previous versions of ASP.NET MVC

Introduction to .NET

.NET SDKs and Runtimes

- .NET Runtime
  - Different version for each platform
  - Provides assembly loading, garbage collection, JIT compilation of IL code, and other runtime services
  - Includes the dotnet tool for launching applications
- ASP.NET Core Runtime
  - Includes additional packages for running ASP.NET Core applications
  - Reduces the number of packages that you need to deploy with your application

Introduction to .NET

.NET SDKs and Runtimes

- .NET SDK
  - Includes the .NET runtime for the platform
  - Additional command-line tools for compiling, testing, and publishing applications
  - Contains everything needed to develop .NET applications (with the help of a text editor)

Introduction to .NET

.NET SDKs and Runtimes

- Each version of .NET has a lifecycle status
  - Current – Includes the latest features and bug fixes but will only be supported for a short time after the next release
  - LTS (Long-Term Support) – Has an extended support period
  - Preview – Not supported for production use
  - Out of support – No longer supported

`dotnet.microsoft.com/download`

Introduction to .NET

Visual Studio

- Visual Studio is available for Windows and macOS
  - Full-featured IDE
- Visual Studio Code is available for Windows, macOS, and Linux
  - Includes IntelliSense and debugging features
  - Thousands of extensions are available for additional functionality

visualstudio.microsoft.com

Introduction to .NET

Visual Studio

- JetBrains also offers an IDE for .NET development called Rider
- Available for Windows, macOS, and Linux
- Includes advanced capabilities in the areas of refactoring, unit testing, and low-level debugging

www.jetbrains.com/rider

What's New in ASP.NET Core 6

What's New in C# 10

- Nullable Reference Types
- Init Only Setters
- Global Using Directives
- File-Scoped Namespace Declarations
- Top-Level Statements
- Record Types
- Changes to the ASP.NET Templates

What's New in C# 10

Nullable Reference Types

- Prior to C# 8, all reference types were nullable
- Most common cause of an application crash is a NullReferenceException
- With nullable reference types, reference types behave more like value types
  - Cannot be null be default
  - Can store null if explicitly requested

```
Person p = null;  // compiler warning

Person? p = null;
```

What's New in C# 10

Nullable Reference Types

- Although introduced in C# 8, nullable reference types were not enabled by default
- Starting with .NET 6, new projects have nullable reference types enabled (via the csproj file)

```
<Nullable>enable</Nullable>
```

- Compiler directives can be used to control the feature anywhere in source code

```
#nullable enable
```

```
#nullable disable
```


What's New in C# 10

Nullable Reference Types

- Null-state analysis tracks the null-state of a reference
- Warnings are generated when code is identified that may dereference a null
- Compiler warnings also generated for types that can be instantiated with uninitialized non-nullable field(s)

What's New in C# 10

Nullable Reference Types

- Various C# language features can be used when working with nullable reference types

```
p1?.GiveRaise();
```

```
p1!.GiveRaise();
```

```
Person p2 = p1 ?? new Person();
```

```
(p1 ??= new Person()).GiveRaise();
```


What's New in C# 10

Init Only Setters

- Init only setters were introduced in C# 9
- Used more extensively in .NET 6

```
public struct WeatherObservation
{
  public DateTime RecordedAt { get; init; }
  public decimal TemperatureInCelsius { get; init; }
  public decimal PressureInMillibars { get; init; }
}
```

What's New in C# 10

Global Using Directives

- In the past, Using directives were only applied in the file where they were defined

```
using System;
```

- The global modifier can now be used to have the directive applied to all source files in the compilation

```
global using System;
```


What's New in C# 10

Global Using Directives

- New .NET 6 projects enable implicit usings by default

```
<ImplicitUsings>enable</ImplicitUsings>
```

- Compiler will generate a collection of global using directives based on your project type
  - C# file in the obj directory

What's New in C# 10

File-Scoped Namespace Declarations

- Namespace declarations previously required scope to be defined via brackets

```
namespace Acme.HR
{
  public class Employee { }
}
```

- C# 10 allows a namespace declaration to define the scope as all code within the file

```
namespace Acme.HR;

public class Employee { }
```


What's New in C# 10

Top-Level Statements

- Every executable .NET assembly must include a Main method to act as the entry point
- C# 9 introduced a feature called top-level statements
- Compiler assumes C# statements defined outside of a type represent the implementation of the Main method
  - Main method itself will be generated by the compiler
- Top-level statements are used by the .NET 6 project templates by default
  - Possible to opt-out (checkbox in Visual Studio)

What's New in C# 10

Record Types

- Record types were introduced in C# 9
- Primarily for supporting immutable data models
- Can be defined using positional parameters

```
public record Person(string FirstName, string LastName);
```

- ... or standard property syntax

```
public record Person
{
    public string FirstName { get; init; } = default!;
    public string LastName { get; init; } = default!;
};
```


What's New in C# 10

Record Types

- Reference type with some of the features of value types
  - Value equality
  - Concise syntax for nondestructive mutation

```
var p2 = p1 with { LastName = "Doe" };
```

- When positional syntax is used:
  - Properties are public init-only
  - Constructor available that matches definition

What's New in C# 10

Record Types

- C# 10 adds support for record structs
- Allows use of record capabilities (e.g., with keyword) with a value type

```
public readonly record struct Point(double X, double Y, double Z);
```

- C# 11 will add the required modifier

```
public record Person(string FirstName, required string LastName);
```


What's New in C# 10

Changes to the ASP.NET Templates

- ASP.NET Core 6 includes new project templates
- Use some (but not all) new C# features
- New model for application initialization

What's New in ASP.NET Core 6

Performance Improvements

- AOT Compilation
- Profile-Guided Optimization
- Arm64 Support
- Hot Reload

Performance Improvements

AOT Compilation

- .NET 6 introduces Crossgen2
- Provides ahead-of-time (AOT) compilation to improves startup time by reducing the amount of JIT compilation required
- Enabled by using the ReadyToRun publish option

```
dotnet publish -c Release -r win-x64 -p:PublishReadyToRun=true
```

- Target platform must be specified
  - Can target any platform if compiling on Windows
  - Cannot target Windows from a non-Windows machine

Performance Improvements

Profile Guided Optimization

- Profile-guided optimization (PGO) is when the JIT compiler generates optimized code based on code paths that are most frequently used
- .NET 6 introduces dynamic PGO
  - Additional optimizations based on injected instrumentation
  - Disabled by default
  - Enabled via an environment variable

```
DOTNET_TieredPGO = 1
```


Performance Improvements

Arm64 Support

- .NET 6 includes support for ...
  - macOS Arm64 ("Apple Silicon")
  - Windows Arm64

Performance Improvements

Hot Reload

- Hot reload allow you to modify an app's source code and instantly apply those changes to the running app
- Available in Visual Studio 2022 and via the dotnet watch command

Performance Improvements

Additional Information

devblogs.microsoft.com/dotnet/performance-improvements-in-net-6

What's New in ASP.NET Core 6

ASP.NET Core Architecture

- Application Initialization
- Minimal APIs
- Blazor Enhancements
- Kestrel and HTTP/3
- HTTP Logging

ASP.NET Core Architecture

Application Initialization

- ASP.NET Core 6 project templates include a streamlined initialization process
- In early versions of ASP.NET Core, WebHost was used to create a builder object

```
WebHost.CreateDefaultBuilder()
```

- Set up some defaults, creates an IWebHostBuilder and is used to create an IWebHost

ASP.NET Core Architecture

Application Initialization

- Startup code was traditionally split between ...
  - Program.cs
    - Application settings
    - Logging
    - HTTP Server
  - Startup.cs
    - Dependency injection
    - Middleware
    - Endpoints (routing)

ASP.NET Core Architecture

Application Initialization

- IWebHostBuilder knows to look for ConfigureServices( ) and Configure( )

```
public class Program
{
  public static void Main(string[] args)
  {
    BuildWebHost(args).Run();
  }

  public static IWebHost BuildWebHost(string[] args) =>
    WebHost.CreateDefaultBuilder(args)
      .UseStartup<Startup>()
      .Build();
}
```

ASP.NET Core Architecture

Application Initialization

- In ASP.NET Core 3, additional workloads were added
  - Worker services, gRPC, and others
- "Generic Host" introduced to act as a common host for all workloads
  - Recommended but not required for traditional web apps
  - ConfigureWebHostDefaults extension method added by ASP.NET to support web-specific concepts

```
public static IHostBuilder CreateHostBuilder(string[] args) =>
  Host.CreateDefaultBuilder(args)
      .ConfigureWebHostDefaults(webBuilder =>
      {
        webBuilder.UseStartup<Startup>();
      };
```


ASP.NET Core Architecture

Application Initialization

- ASP.NET Core 6 introduces the WebApplication and WebApplicationBuilder types
- When combined with new C# language features like top-level statements and implicit using directives, the result is a much cleaner configuration experience

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddRazorPages();

var app = builder.Build();

app.UseStaticFiles();
app.MapRazorPages();

app.Run();
```

ASP.NET Core Architecture

Application Initialization

- The generic host is still used
- WebApplication and WebApplicationBuilder act as wrappers and provide a simpler API
- WebApplicationBuilder is used to ...
  - Load the application configuration
  - Add services
  - Configure logging
  - Everything else except middleware

ASP.NET Core Architecture

Application Initialization

- Once configured, the Build( ) method of WebApplicationBuilder is used to create and instance of WebApplication

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

- The WebApplication object is used to configure middleware and endpoints
- Run( ) method of WebApplication used to start the application

ASP.NET Core Architecture

Minimal APIs

- Minimal APIs is a new sub-framework that can be used to create services with a minimum of files, features, and dependencies
- Alternative to a "controller-based web API"
- Good fit for simpler, smaller services (microservices)

```
app.MapGet("/todoitems", async (TodoDb db) =>
  await db.Todos.ToListAsync());

app.MapGet("/todoitems/complete", async (TodoDb db) =>
  await db.Todos.Where(t => t.IsComplete).ToListAsync());

app.MapGet("/todoitems/{id}", async (int id, TodoDb db) =>
  await db.Todos.FindAsync(id)
    is Todo todo
      ? Results.Ok(todo)
      : Results.NotFound());
```


ASP.NET Core Architecture

Blazor Enhancements

- _Layout.cshtml used for layout instead of _Host.cshtml
- AOT compilation available for Blazor WebAssembly apps
- Error boundaries

```
<ErrorBoundary>
  <ChildContent>
    @Body
  </ChildContent>
  <ErrorContent>
    <p class="errorUI">Nothing to see here right now. Sorry!</p>
  </ErrorContent>
</ErrorBoundary>
```

- Blazor Hybrid apps with .NET MAUI

ASP.NET Core Architecture

Kestrel and HTTP/3

- HTTP/3 is available in .NET 6 as a preview feature
- Not enabled by default

```
builder.WebHost.ConfigureKestrel((context, options) =>
{
  options.ListenAnyIP(5001, listenOptions =>
  {
    listenOptions.Protocols = HttpProtocols.Http1AndHttp2AndHttp3;
    listenOptions.UseHttps();
  });
});
```

- HTTP/3 will be silently unavailable if the underlying OS does not support it

ASP.NET Core Architecture

Kestrel and HTTP/3

- HTTP/3 is advertised by the server via the alt-svc header
  - Client can then establish an HTTP/3 connection if desired
- HTTP/3 does not allow the use of self-signed certificates
- Benefits include ...
  - Faster response time for initial request
  - Better handling of packet loss
  - Improved support for network switching (e.g., WiFi to cellular)

ASP.NET Core Architecture

HTTP Logging

- HTTP logging is new built-in middleware that logs information about HTTP requests and responses

```
var builder = WebApplication.CreateBuilder(args);

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

app.MapGet("/", () => "Hello World!");

app.Run();
```


ASP.NET Core Architecture

HTTP Logging

- Can be configured via HttpLoggingOptions

```
builder.Services.AddHttpLogging(logging =>
{
  logging.LoggingFields = HttpLoggingFields.All;
  logging.RequestBodyLogLimit = 4096;
  logging.ResponseBodyLogLimit = 4096;
});
```

What's New in ASP.NET Core 6

Migrating an Existing Application to .NET 6

- .NET 5
- Earlier Versions of .NET Core
- .NET Framework
- .NET Upgrade Assistant

Migration

From .NET 5

- It should be possible to move an existing ASP.NET Core 5 application to .NET 6 without any code changes

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
  </PropertyGroup>
</Project>
```

- Update any other package references to the appropriate version
- If using Docker, update your Dockerfile to use the correct base image

Migration

From .NET 5

- Changes that you should consider ...
  - Use of new language features such as top-level statements, global/implicit usings, file-scoped namespaces
  - Modernization of application initialization (WebApplicationBuilder)
  - Nullable reference types

Migration

From .NET Core

- Migrating from .NET Core 3.x should be very similar to the process for migrating from .NET 5
- For earlier versions of .NET Core, consult the official documentation for guidance
  - For example, migrating from Core 1.1 may require multiple steps (1.1 -> 2.0 -> 3.1 -> 6)

Migration

From .NET Framework

- Class library projects should require minimal changes
  - .NET Standard libraries can be used as-is
  - Windows-specific code will not work on a non-Windows platform
- ASP.NET projects with Razor views (ASP.NET MVC) should be reusable as-is (including view models)
- Web Forms (.aspx) will need to be rewritten although some web artifacts may be reusable (JavaScript, CSS, etc.)

Migration

.NET Upgrade Assistant

- The .NET Upgrade Assistant is a command-line tool that can be run on different kinds of .NET Framework apps
- The Platform compatibility analyzer tries to identify if you are using an API that will throw a PlatformNotSupportedException