



Session 7

Improvements in ASP.NET Core

Session Overview

- Describe .NET Core MVC and Razor page improvement
- Explain MVC Model Binding improvements
- Describe Text.Json
- Describe dump debugging
- Outline Performance improvements in recent versions
- Identify the use of Garbage Collection

.NET Core MVC and Razor Pages (1-3)

1

- Razor Pages is an engine that creates views of MVC design pattern in Microsoft Visual Studio.

2

- Razor uses a markup syntax to embed server-based code in languages such as Visual Basic and C# into Web pages.

3

- Razor view engine has support of both ASP.NET MVC and Core MVC.

Characteristics of Razor syntax:

Compact

- Enables to write easy and simple codes by minimizing the number of characters and keystrokes

Easy to Learn

- Enables use of languages such as C# or VB

IntelliSense

- Includes support within Visual Studio in Razor syntax for statement completion

.NET Core MVC and Razor Pages (2-3)

Razor-based MVC Application

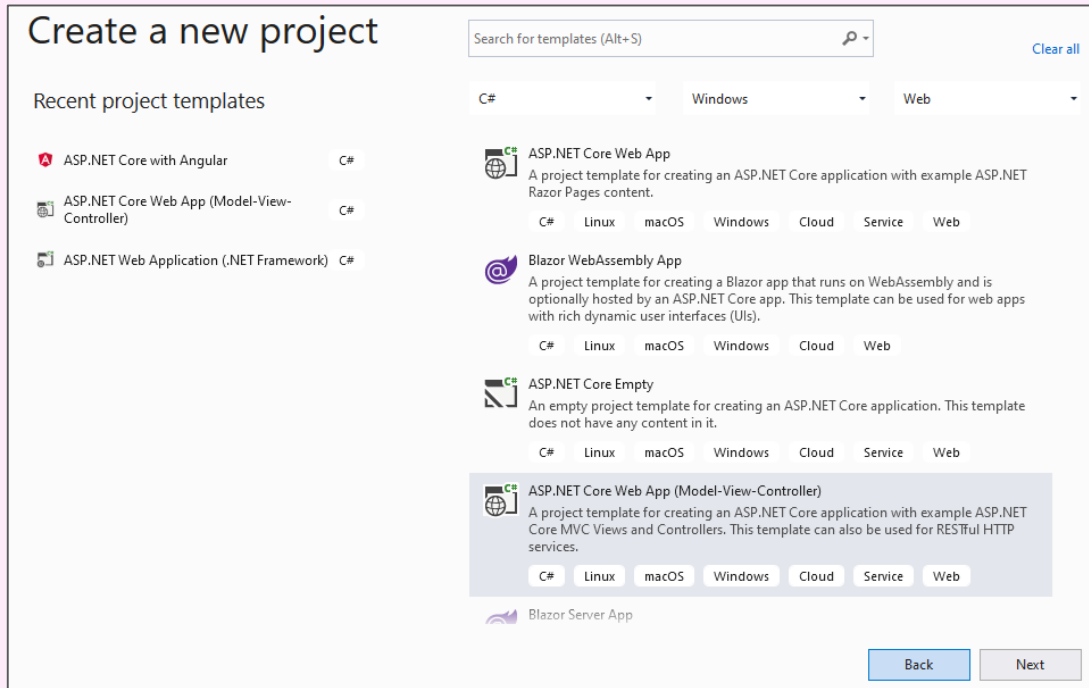


Figure 7.1: Creating a New Project

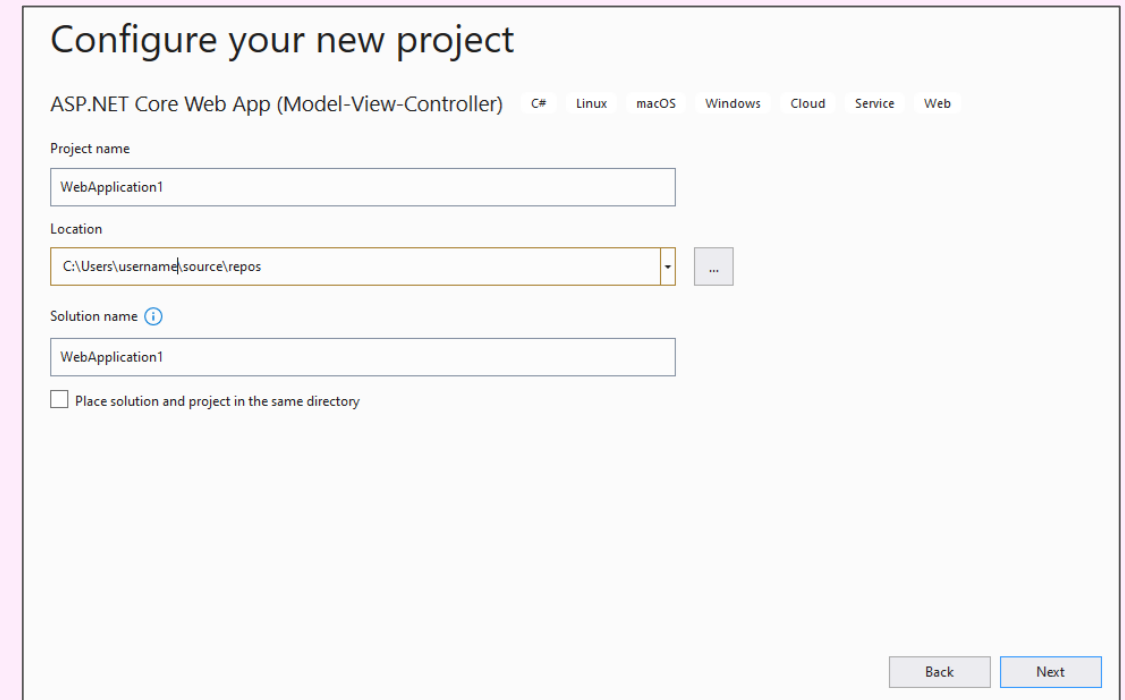
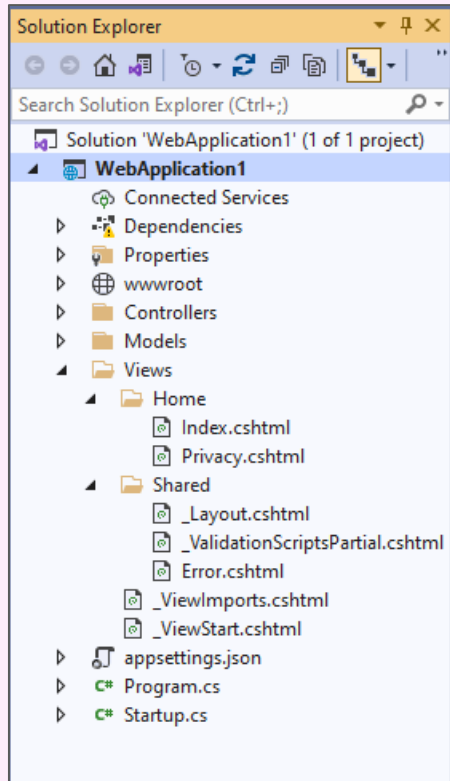
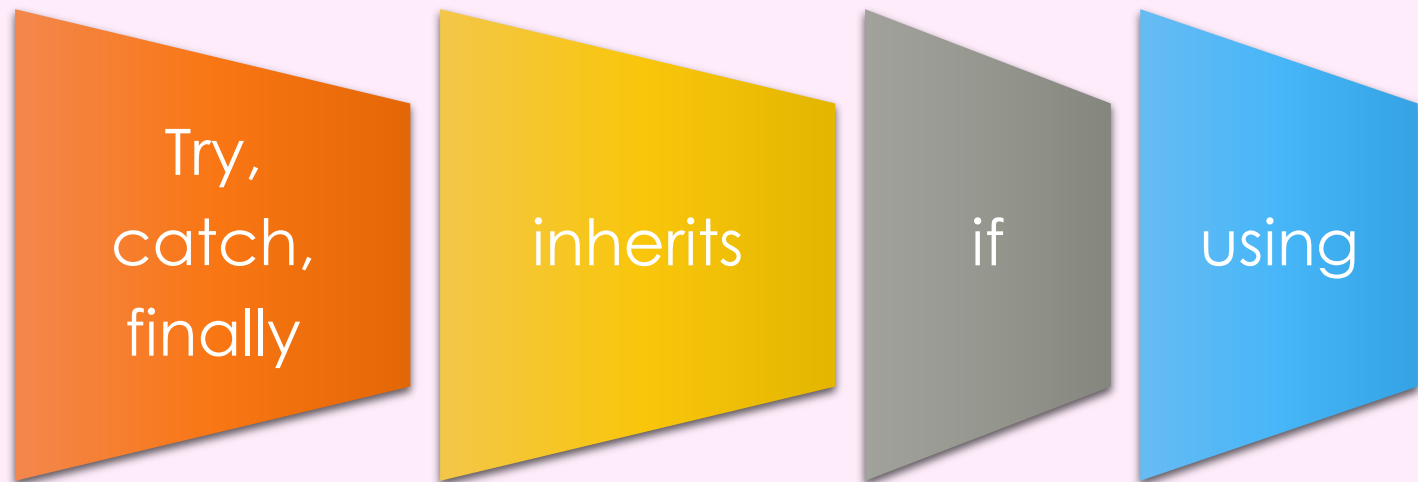


Figure: 7.2: Configure New Project

.NET Core MVC and Razor Pages (3-3)



**Figure 7.3: Directory
Structure in Solution Explorer**



**Most frequently used
Razor commands.**

.MVC Model Binding

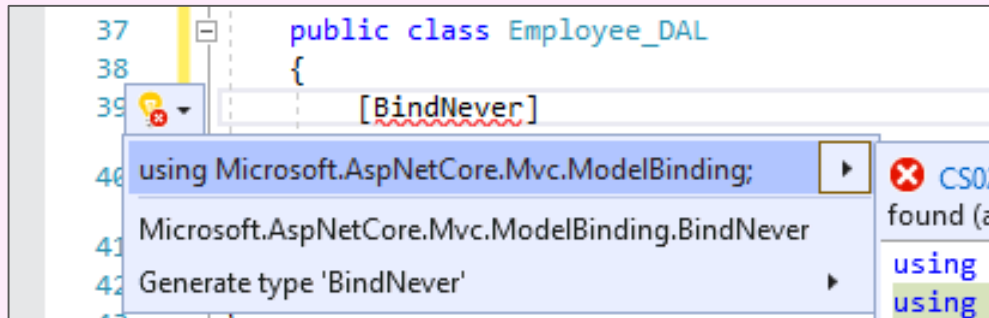
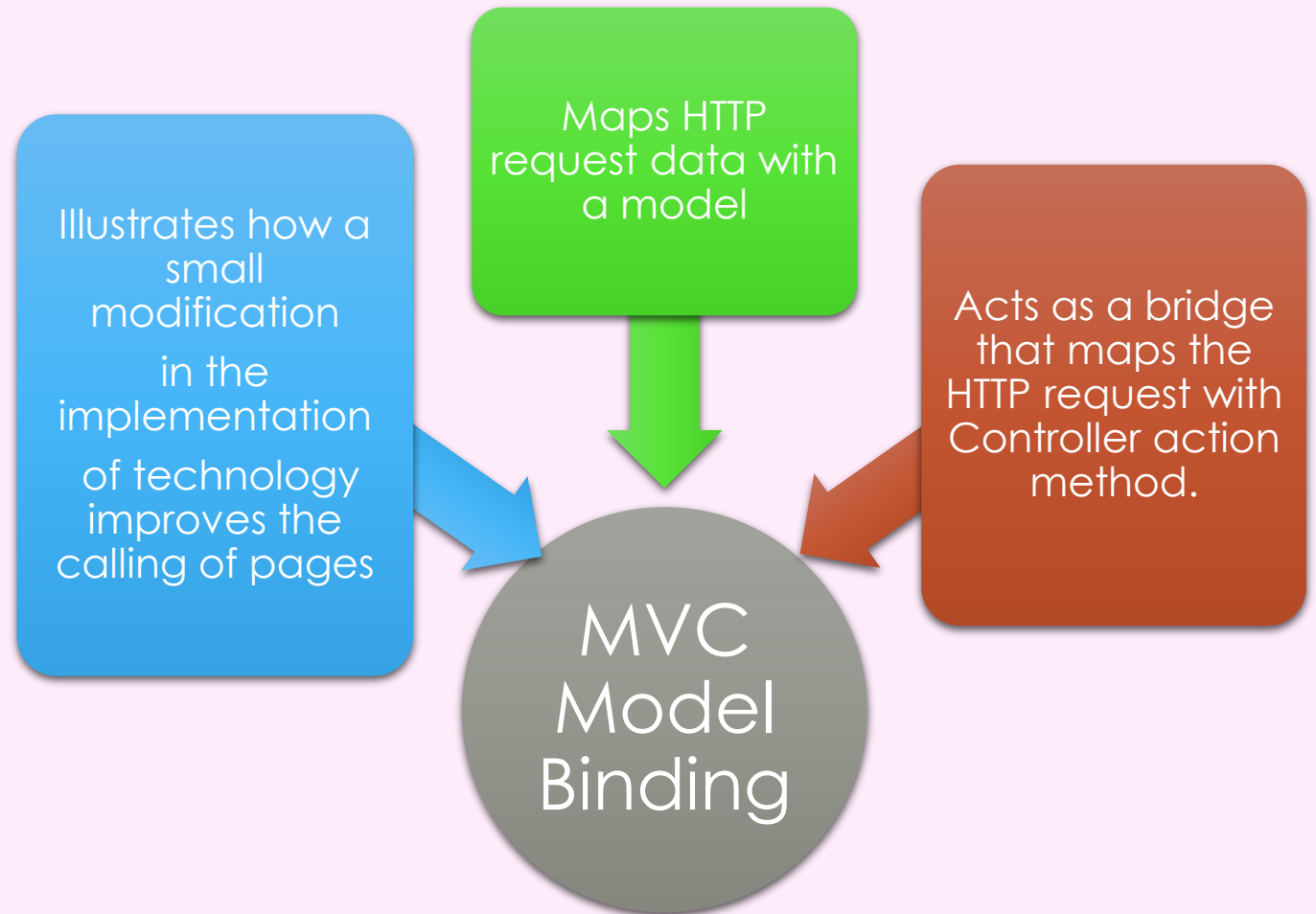
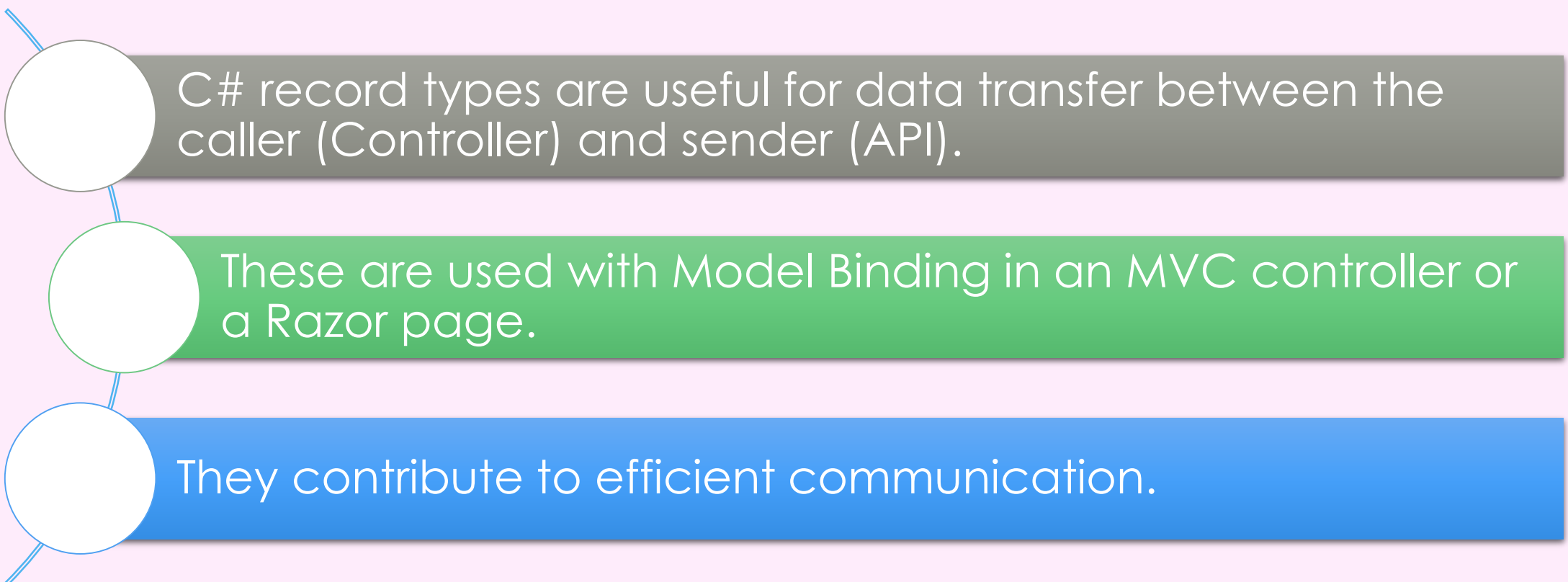


Figure 7.5: Model Binding



C#9 Record Type



C# record types are useful for data transfer between the caller (Controller) and sender (API).

These are used with Model Binding in an MVC controller or a Razor page.

They contribute to efficient communication.

Text.Json

JSON Objects:

JsonSerializer

Allows serialization and deserialization of .NET objects to and from JSON.

JsonDocument

Allows for better structural content management rather than data values being instantiated.

JsonElement

Refers to a JSON value in a JsonDocument.

Utf8JsonWriter

Provides API for writing UTF-8 encoded JSON content and writes the text with no-cache.

Utf8JsonReader

Provides API for reading UTF-8 encoded JSON content and reads text without caching.

Features of JsonSerializer:

Serialization and Deserialization:

Provides support for serializing and deserializing objects.

UTF 8 Data:

Allows native processing of UTF-8 data.

JsonNamingPolicy

Specifies custom naming policies.

Asynchronous support:

Built-in support for asynchronous serialization and deserialization.

Dump Debugging

**Dotnet-dump
collect:**

- Collects the dump based on name, process id, and other parameters.

**Dotnet-dump
analyze:**

- Explores and analyzes the dump.

Runtime Libraries

Runtime libraries are base of all .NET class.

These libraries have NuGet packages as their extensions.

To improve performance, runtime libraries should be annotated for nullable reference types.

Garbage Collector

Automatically manages memory in the .NET framework.

Manages memory allocation and the lifetime of objects.

Reduces application downtime.

Determines the time for garbage collection.

Performs memory management through managed heaps.

Summary

- Razor is a popular view-engine or a markup language used at the server-side.
- ASP.NET MVC supports the Razor view engine.
- Both HTML and server-side code using C# or Visual Basic (VB) are employed for writing Razor syntax.
- For an object to be immutable and behave like a value, it should be declared as a record.
- JSON supports UTF-8 text encoding and serialization library for converting .NET object types to a JSON string or vice versa.
- Any class libraries used by a project must be annotated for nullable reference types.
- In ASP.NET Core 5.0, the runtime libraries are 80% annotated for nullable reference types.
- The garbage collector is an automatic memory manager that maintains the lifetime of objects.