New Features In .NET 8

Introduction

Today, we will look into some new features expected in .NET 8. .NET 8 will be the next version of the .NET framework after .NET 7. It will be an LTS (Long-term support) release.

How to use .NET 8?

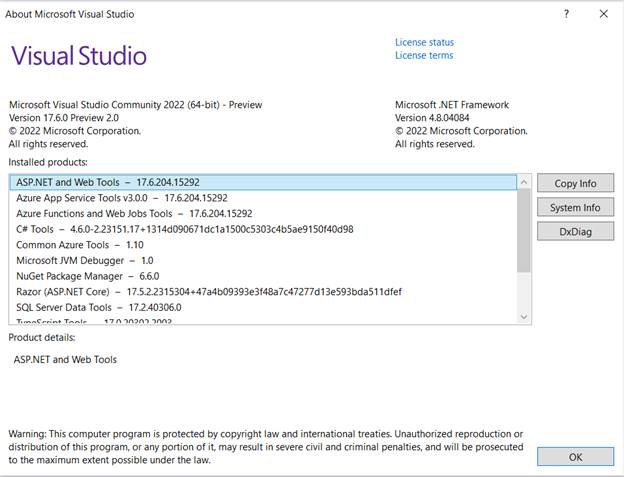
The .NET 8 SDK can be downloaded from the below URL,

<https://dotnet.microsoft.com/en-us/download/dotnet/8.0>

Kindly note that to use .NET 8 in Visual Studio, we need to have a version of Visual Studio 2022 (v17.6 Preview 1) or a later version.

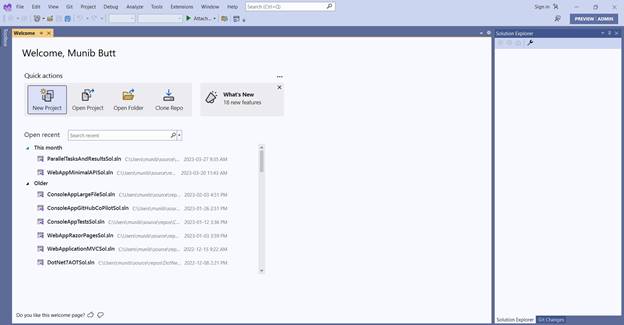
Creating the project and code

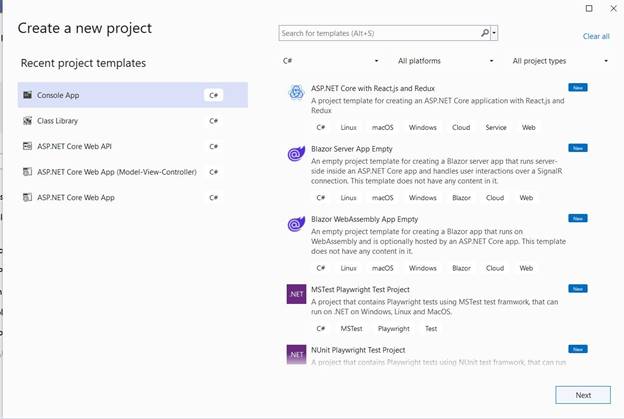
I will create this application using Visual Studio 2022 Community Version 17.6.0 Preview 2.0 edition.

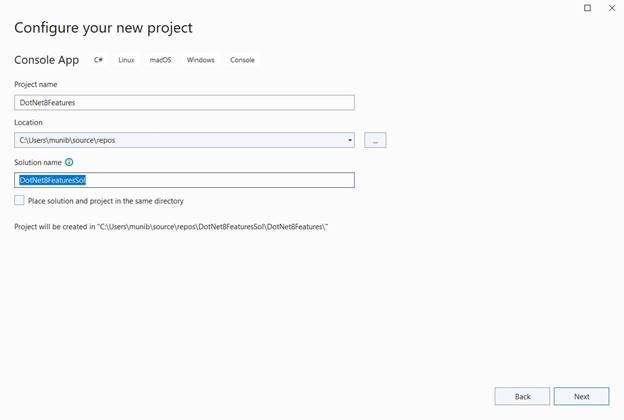


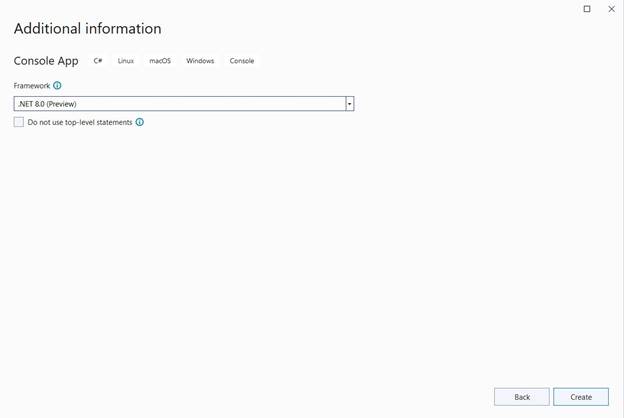
Please follow the below steps.

Create a C# console application.

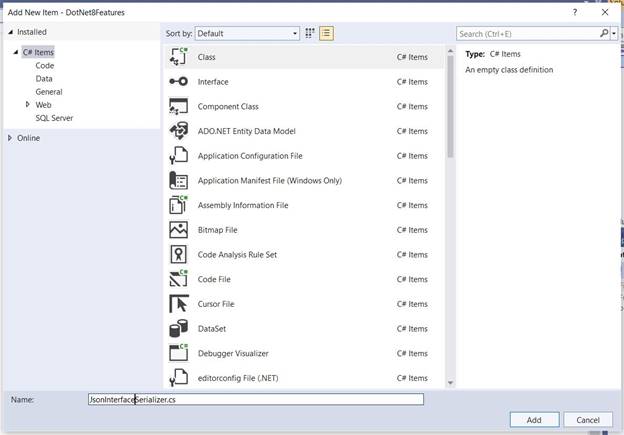




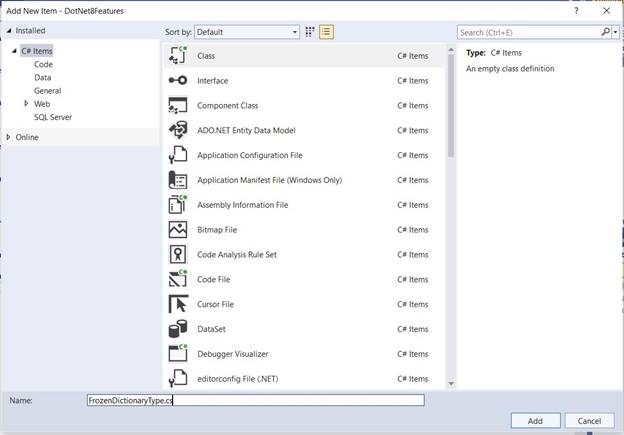




Next, add a new class as below,



Then, add another class as below,



Finally, add the below code in the respective files:

// FrozenDictionaryType.cs

using System.Collections.Frozen;

namespace DotNet8Features {

internal class FrozenDictionaryType {

private static readonly FrozenDictionary < string, string > staticData = LoadStaticData().ToFrozenDictionary();

private static Dictionary < string, string > LoadStaticData() {

return new Dictionary < string, string > () {

{

"Key1",

"Value1"

}, {

"Key2",

"Value2"

}, {

"Key3",

"Value3"

}

};

}

internal string GetValue(string key) {

if (staticData.TryGetValue(key, out string ? keyValue)) {

return keyValue;

}

return string.Empty;

}

}

}

// JsonInterfaceSerializer.cs

namespace DotNet8Features {

public interface ICar {

public string ? Brand {

get;

set;

}

}

public interface ISedan: ICar {

public string ? Model {

get;

set;

}

}

public class CarDetails: ISedan {

public string ? Brand {

get;

set;

}

public string ? Model {

get;

set;

}

}

}

// Program.cs

using DotNet8Features;

using System.Text.Json;

// Feature 1: Interface serialization

ISedan detail = new CarDetails {

Brand = "Honda", Model = "Civic"

};

var serializedValue = JsonSerializer.Serialize(detail);

Console.WriteLine(serializedValue);

// Feature 2: Frozen dictionaries

FrozenDictionaryType frozenDictionaryType = new();

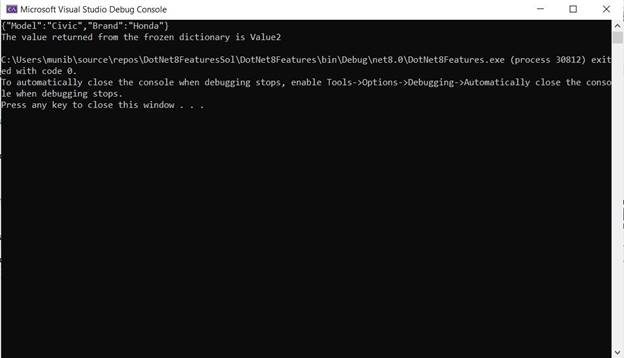
Console.WriteLine($ "The value returned from the frozen dictionary is {frozenDictionaryType.GetValue("

Key2 ")}");

C#

Copy

Now, when we run our application, we see the below,



Here, we have used two features of .NET 8.

The first feature is the ability to serialize properties from interface hierarchies. We see this feature implemented in the JsonInterfaceSerializer class. Here we created a base interface called ICar and a sub-interface called ISedan. We then created a class instance implementing ISedan and assigned values to both properties in the immediate and parent interfaces. Both these values were serialized using the “System.Text.Json” package.

The second feature is the new performance-focused type “FrozenDictionary”. This is for improved read operations. This type does not allow changes to keys and values in the dictionary after creation.

The code is available at the below location,

<https://github.com/munibrbutt/articles-code/tree/main/Some%20.NET%208%20features>

