INF 272
Advanced C#

24/02/2025

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

Session 3: Working with ASP.NET MVC (Roadmap)

- What is MVC?
- The Pattern Composition
- The Code Environment
  - Starting a new MVC project
  - Structure of a MVC project
  - Adding new components
  - Simple MVC project
- Practical overview

Today's Session

1. Brief administrative overview
2. Adjustment made to the study guide
3. Position on class attendance
4. Commencement of practicals
5. Tutorial sessions

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

Adjustment to the study guide

2.3.1 Theory and practical preparation sessions Page 6.

A student will be required to attend and participate in **ONE theory session a week**. In the listed sessions students will be introduced to concept, programming syntax, examples etc. as well as the brief, discussion and description of the week's subsequent practical sessions. **Attendance of the theory sessions, and at least one practical session per week is compulsory**. **Attendance of the theory sessions shall be taken randomly during the module. Students with high attendance records will get a bonus (between 3 - 5 marks) which will be added to their final module mark. Also, there will be regular impromptu in-class exercises (quizzes) during the theory sessions that will count towards the final module mark of each student**. **The details in these theory sessions will not be repeated in the practical sessions.** Hence, all students are advised to take participation in the theory sessions seriously. There will be no remedy for marks missed due to non-participation in theory sessions.

4.1 Final semester mark event contributions and notes Page 14.

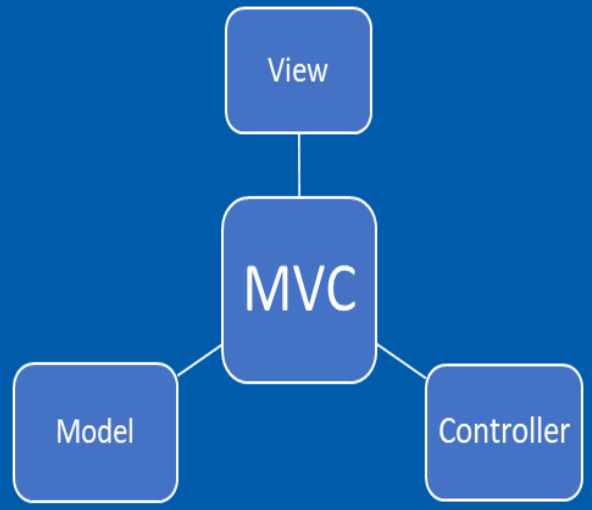
Module Average [as calculated at the end of the year]	
• Practical activities and tasks (25%) ; in-class exercises (5%)	30%
• Projects and homework contribution	30%
• Semester Test 1 (S1) and Semester Test 2 (S2) contribution	40%
• **Bonus marks**: Attendance (100%: **5**; 80-90%: **4**; 60-70%: **3**); Class engagement	TBD
Final year average [40% required to write exam]	**100%**

Working with ASP.NET MVC

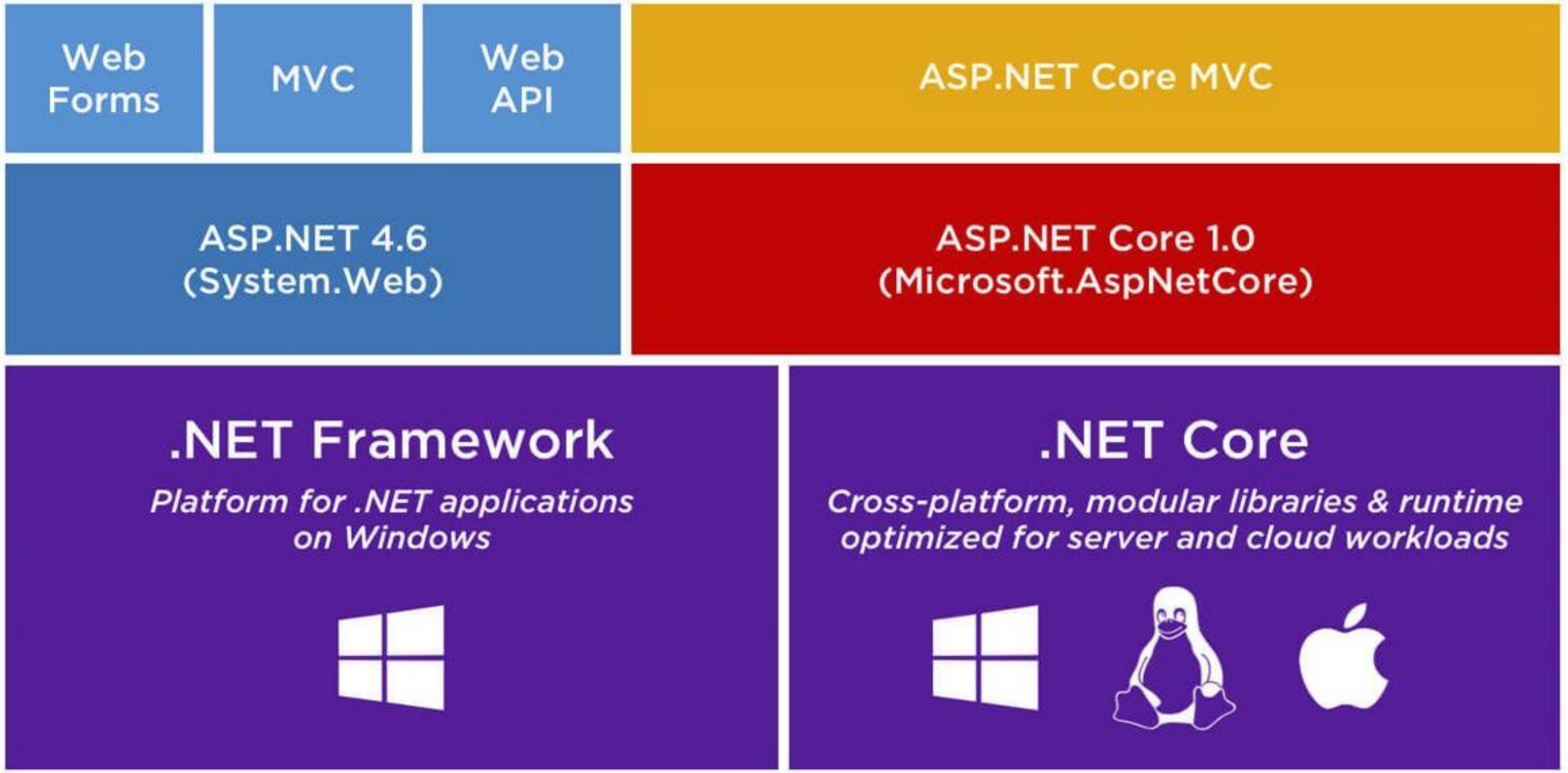
What is MVC?

MVC stands for Model-View-Controller.

A popular software design pattern used for developing web applications and user interfaces.



ASP.NET MVC


Web Forms
MVC
Web API
ASP.NET Core MVC
ASP.NET 4.6 (System.Web)
ASP.NET Core 1.0 (Microsoft.AspNetCore)
.NET Framework
Platform for .NET applications on Windows
.NET Core
Cross-platform, modular libraries & runtime optimized for server and cloud workloads

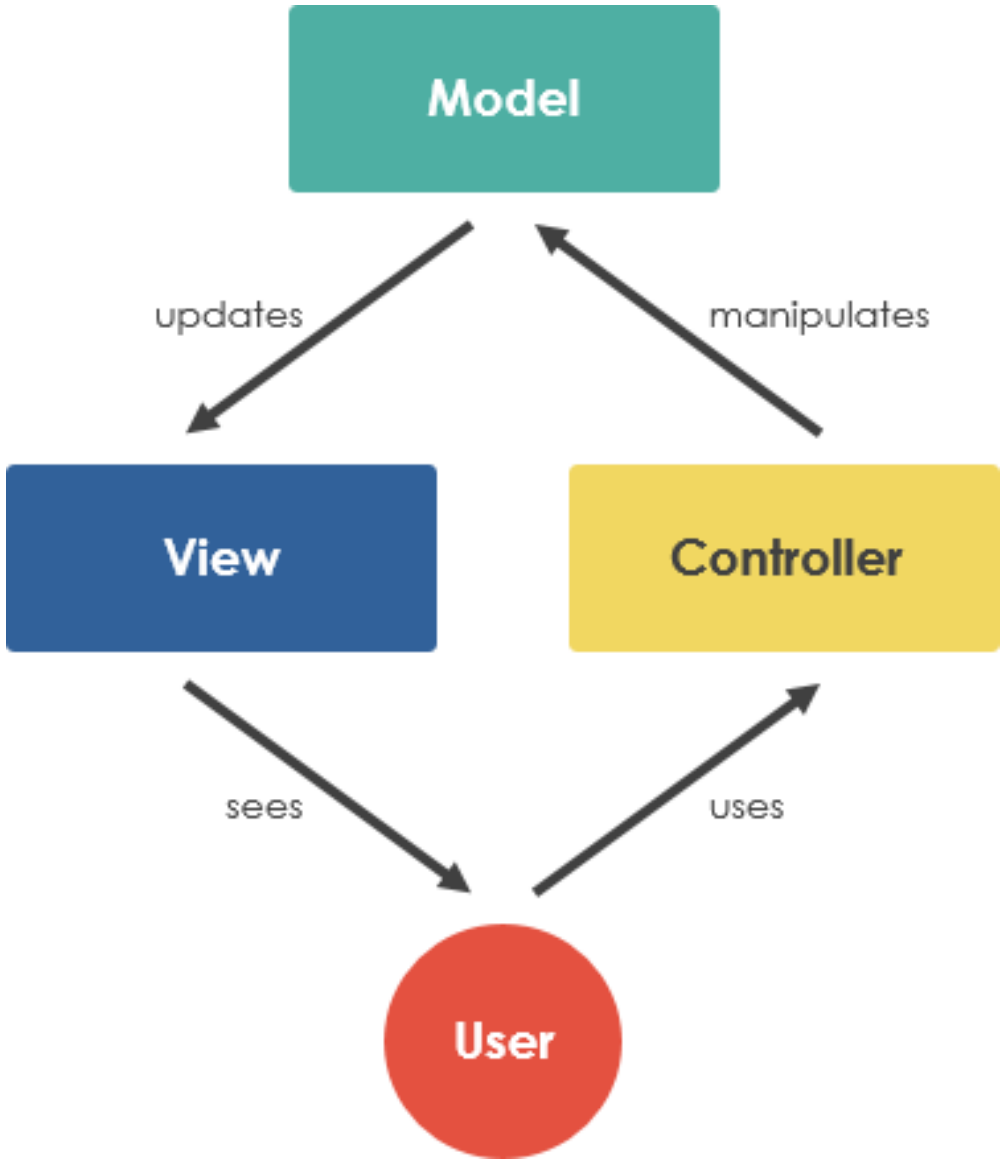
To summarize:

- MVC is a software pattern to create web application using ASP.NET
- ASP.NET is seen as a Web Application Framework within the broader .NET Architecture Ecosystem

Working with ASP.NET MVC

The Pattern Composition

Basic MVC Diagram



- User sees the View
- User makes requests via the Controller
- Controller manipulates the Model
- Model updates the View

Model

- Represents the core data and the business logic of your application
- Manages application data (retrieving, storing, updating)
- Encapsulates business rules and logic that govern how the data should be manipulated or changed
- Often interacts with databases or external services to persist data
- Notifies the View about any changes in its state

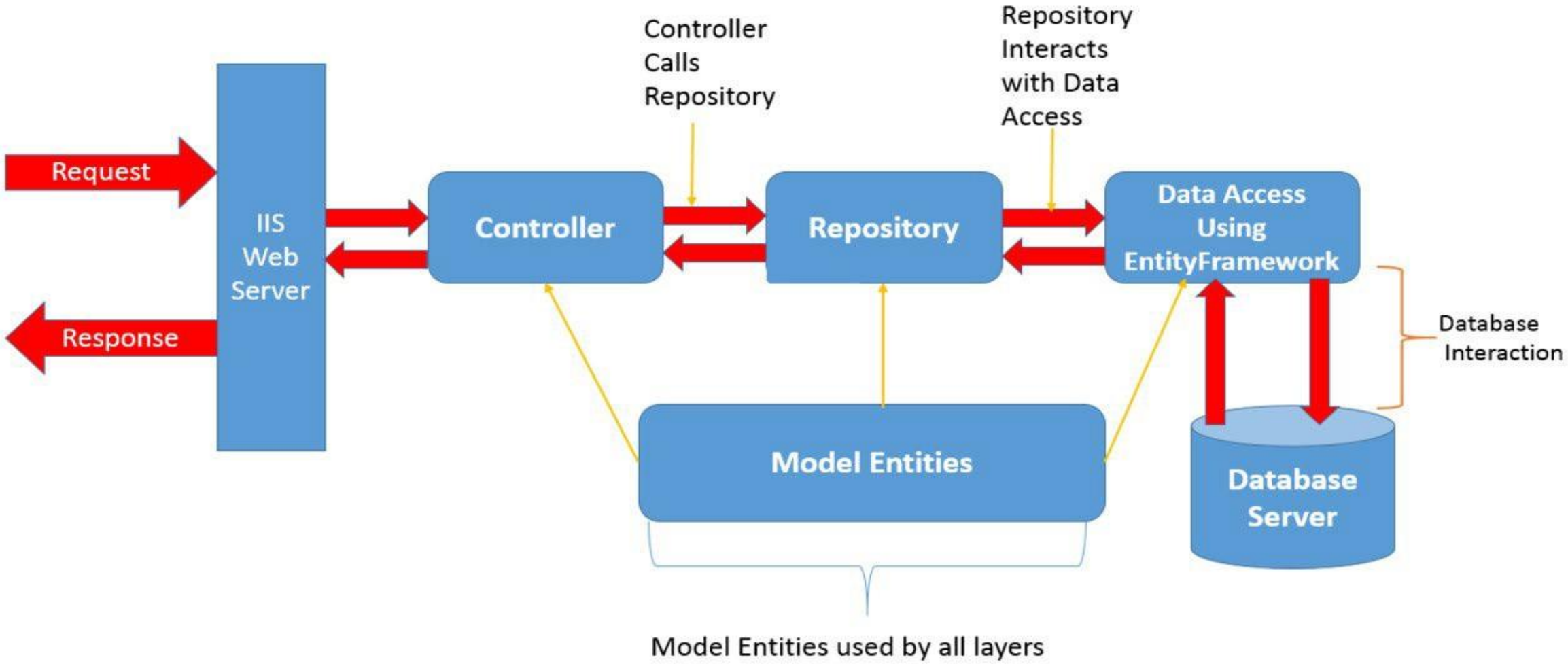
View

- Responsible for how the information is presented to the user
- Renders the UI using data provided by the Model
- Displays information in a visually appealing and user-friendly way
- Handles user interactions such as button clicks, form submissions, etc.
- Sends user input to the Controller for processing
- Receives updates from the Controller
- Represents the presentation layer of the application
- It's the user interface for interacting with the application

Controller

- Receives and processes user input as well as handling system events
- Interacts with the model to perform business operations and data manipulation
- Selects the appropriate View to display to the user based on the results of the interaction
- Updates the view based on changes in the model's data
- User actions trigger the Controller, which updates the Model if necessary, and the Model then usually notifies the View that a change has occurred
- May contain application-specific logic related to user interactions and application behavior
- **Acts as a bridge between the Model and the View**

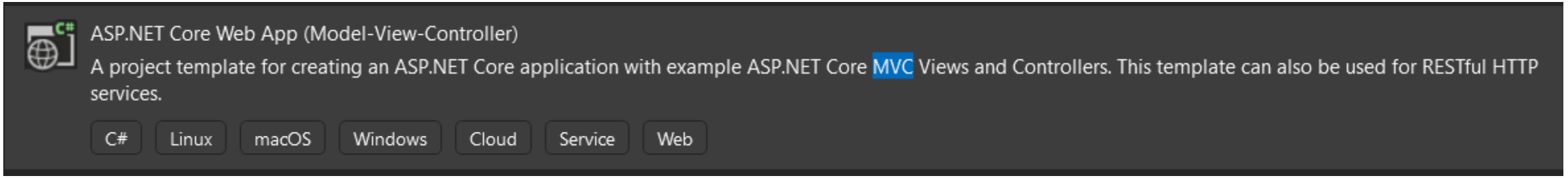
Another MVC Diagram

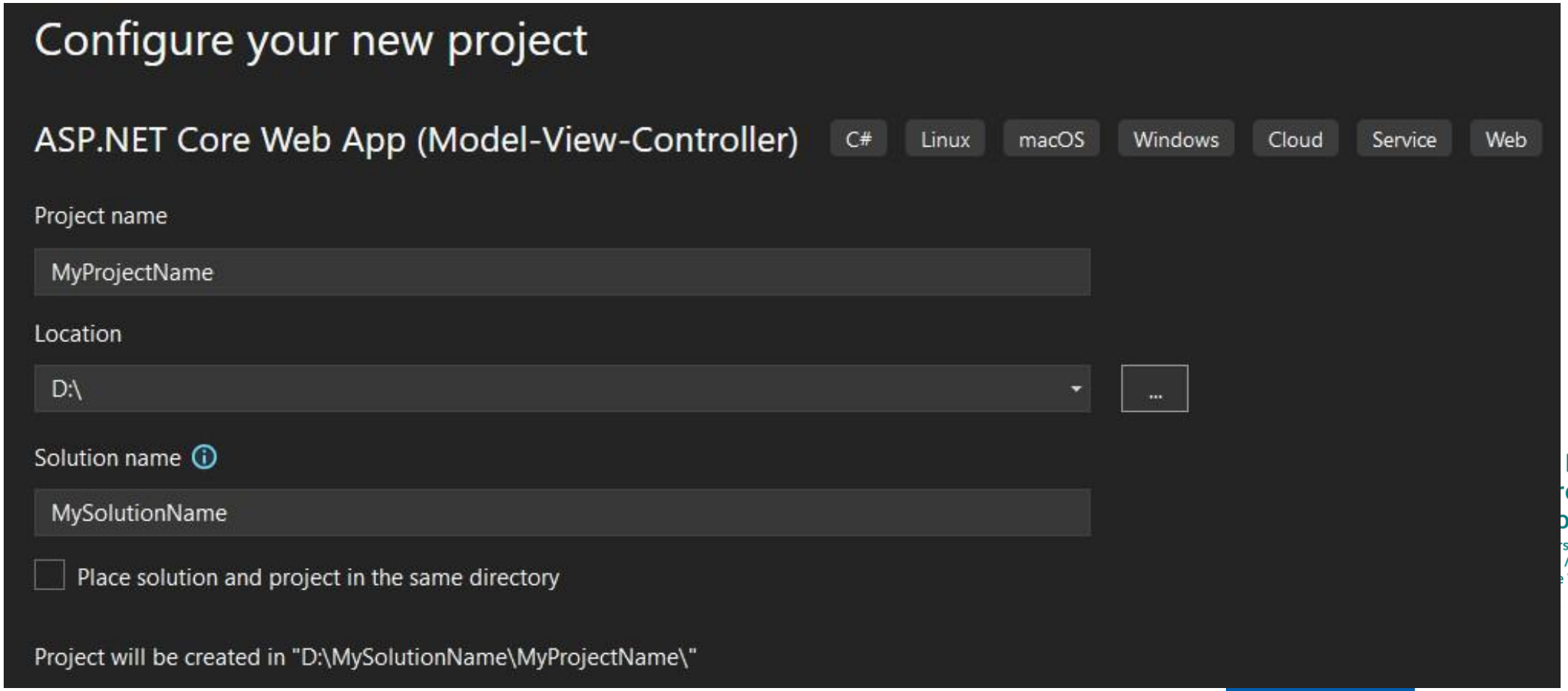


Working with ASP.NET MVC

The Code Environment

Basic MVC Environment





Engineering,
ronment and
on Technology
rswese, Bou-omgewing en
/ Lefapha la Boetšenere,
e Theknolotši ya Tshedimošo

Create a new project

Search for templates (Alt+S)

Clear all

C# | Windows | All project types

Recent project templates

ASP.NET Web Application (.NET Framework) C#

Select ASP.NET Application. Make sure it is for MCV for C#, Windows

Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows Console

ore Web App
template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content
inux macOS Windows Cloud Service Web

Blazor Server App
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Blazor Cloud Web

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.
C# Linux macOS Windows Cloud Service Web WebAPI

Class Library
A project for creating a class library that targets .NET or .NET Standard
C# Android Linux macOS Windows Library

ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
C# Linux macOS Windows Cloud Service Web

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.

Next

Configure your new project

ASP.NET Web Application (.NET Framework)

C# Windows Cloud Web

Project name

Sample

Location

C:\Users\User\source\repos

...

Solution

Create new solution

Solution name

Sample

Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Project will be created in "C:\Users\User\source\repos\Sample\Sample\"

Provide an appropriate name that should not start with any numbers or contain spaces inside the name.

Back Create

Create a new ASP.NET Web Application

Empty

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

Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication

None

Add folders & core references

- [ ] Web Forms
- [x] MVC
- [ ] Web API

Advanced

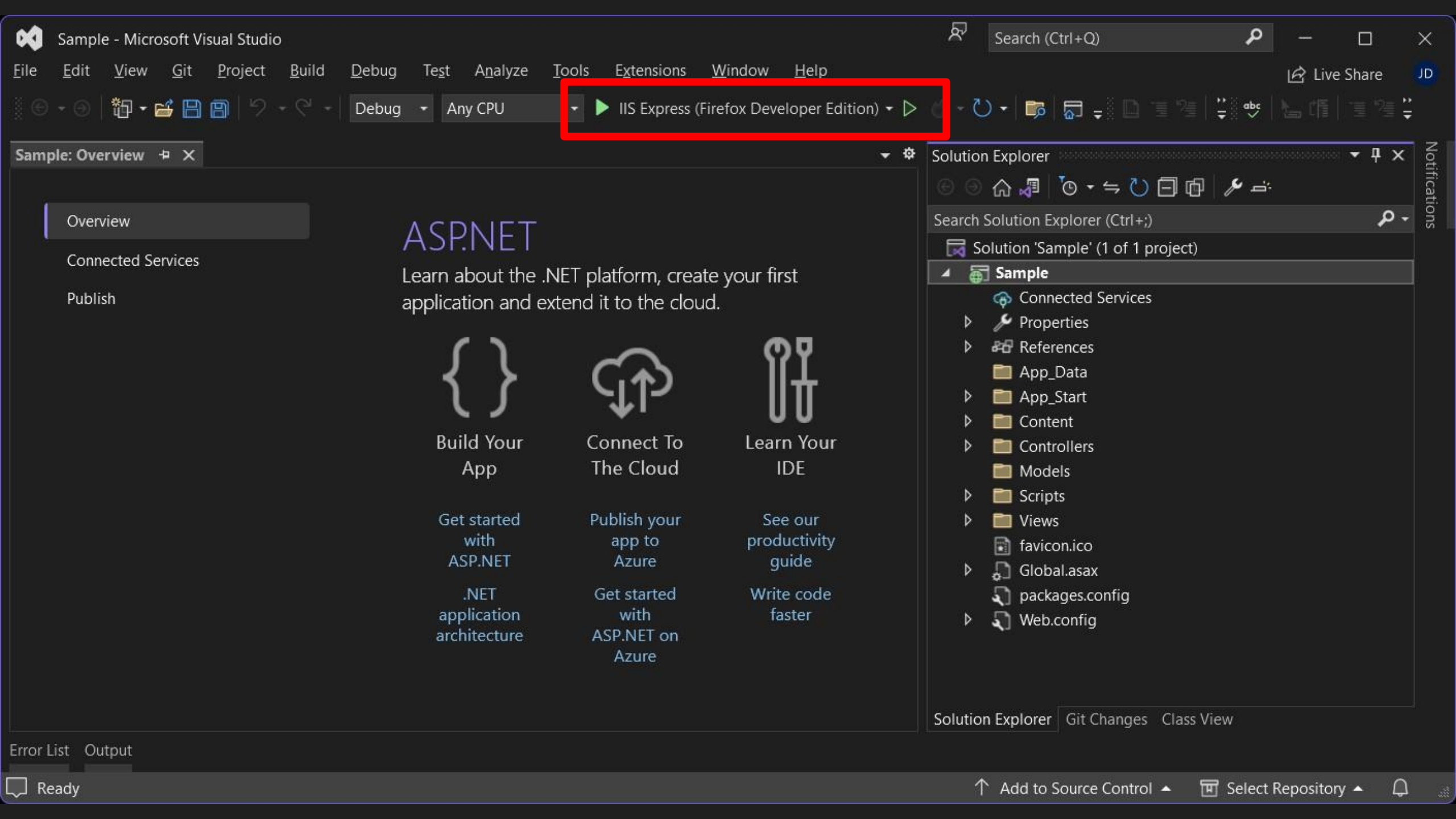
- [ ] Configure for HTTPS
- [ ] Docker support

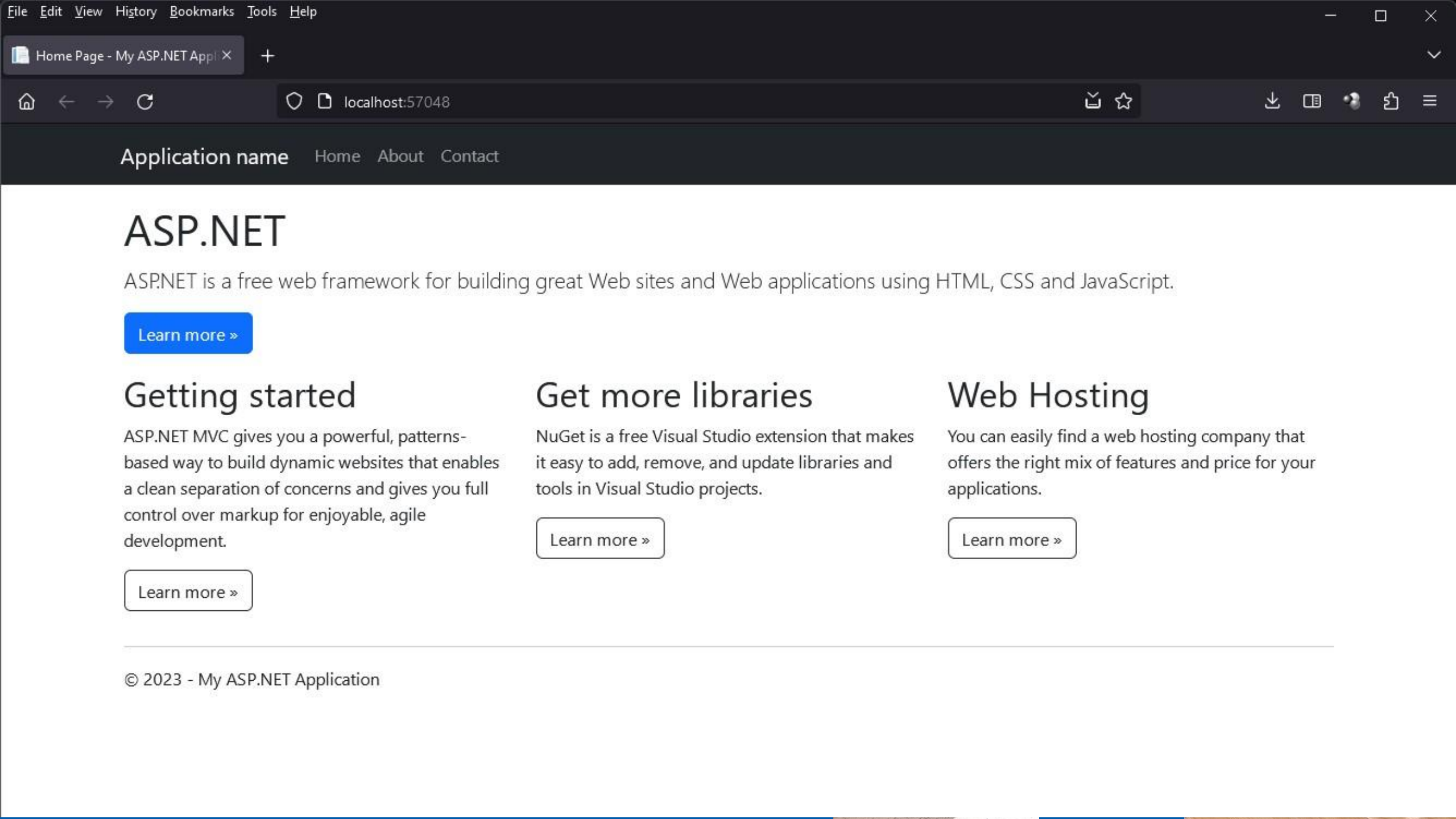
(Requires Docker Desktop)

- [ ] Also create a project for unit tests

Sample.Tests

Back Create


Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
JD
Debug
Any CPU
IIS Express (Firefox Developer Edition)
Sample: Overview
Overview
Connected Services
Publish
ASP.NET
Learn about the .NET platform, create your first application and extend it to the cloud.
Build Your App
Get started with ASP.NET
.NET application architecture
Connect To The Cloud
Publish your app to Azure
Get started with ASP.NET on Azure
Learn Your IDE
See our productivity guide
Write code faster
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Sample' (1 of 1 project)
Sample
Connected Services
Properties
References
App_Data
App_Start
Content
Controllers
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
Class View
Notifications
Error List
Output
Ready
Add to Source Control
Select Repository


File Edit View History Bookmarks Tools Help
Home Page - My ASP.NET Appl
localhost:57048
Application name
Home
About
Contact
ASP.NET
ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.
Learn more »
Getting started
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.
Learn more »
Get more libraries
NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.
Learn more »
Web Hosting
You can easily find a web hosting company that offers the right mix of features and price for your applications.
Learn more »
© 2023 - My ASP.NET Application

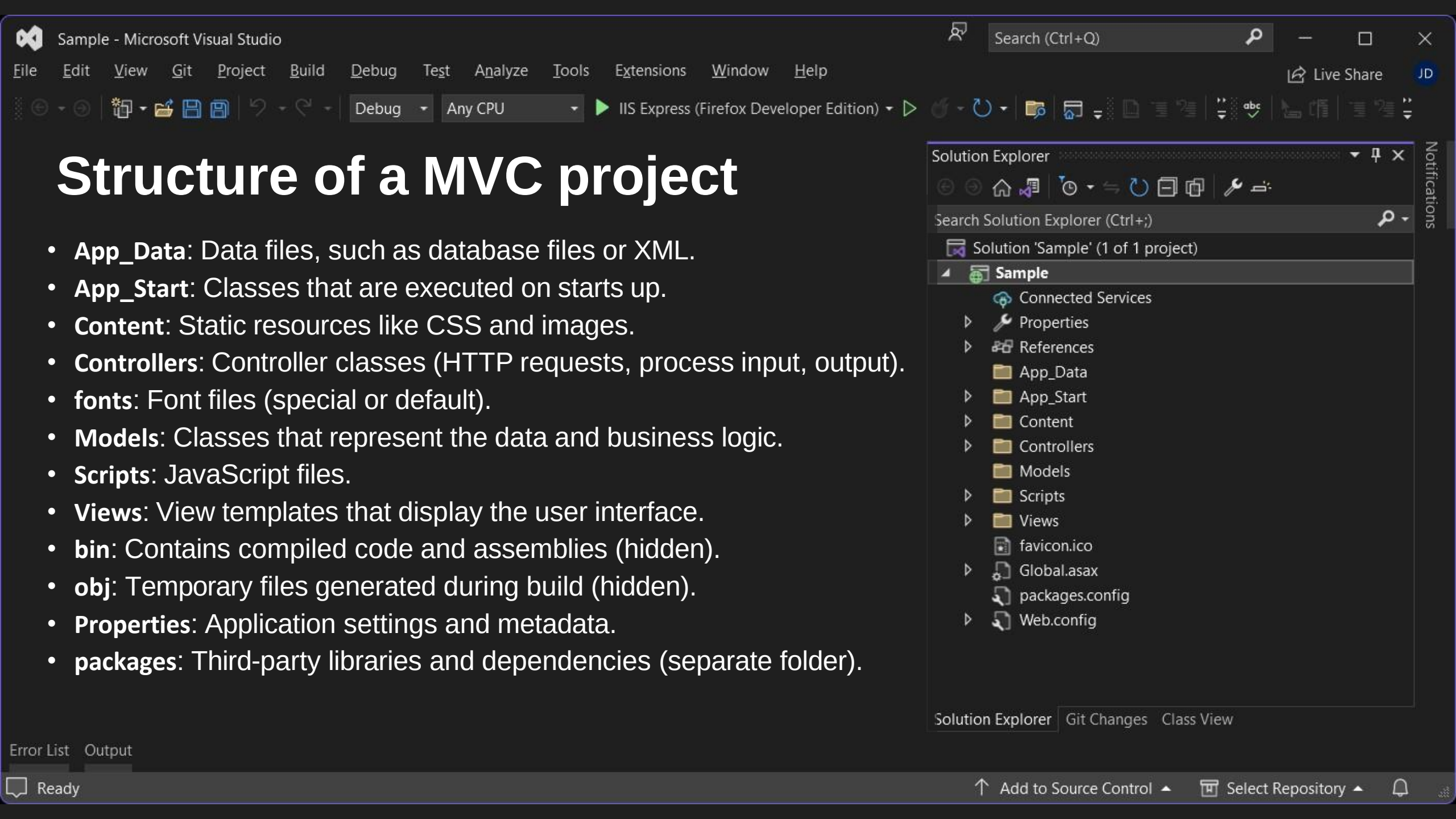
Structure of a MVC project

A brief overview of the main folders and details.

There are additional details that will not be dived into in this session.

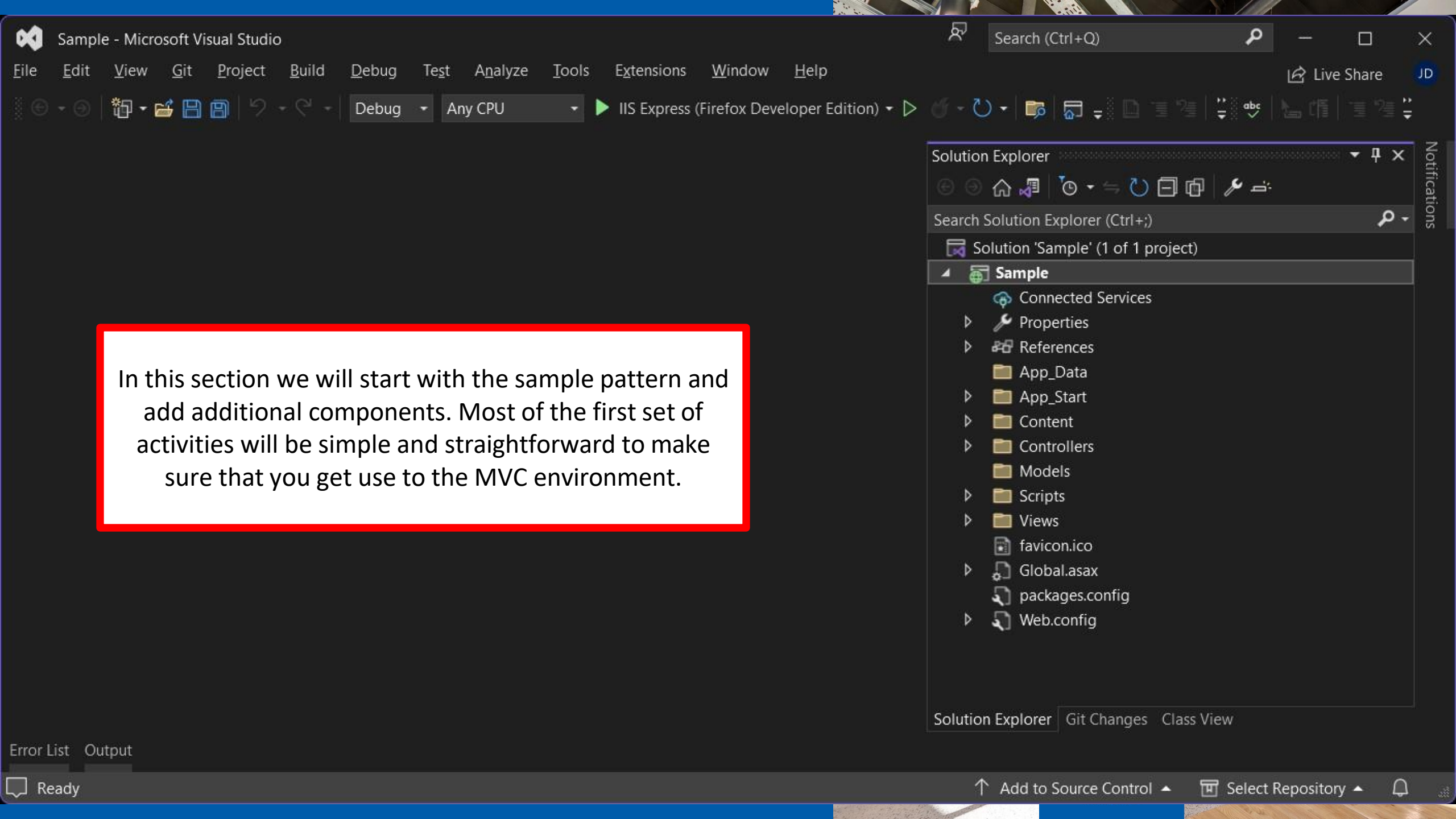
Structure of a MVC project

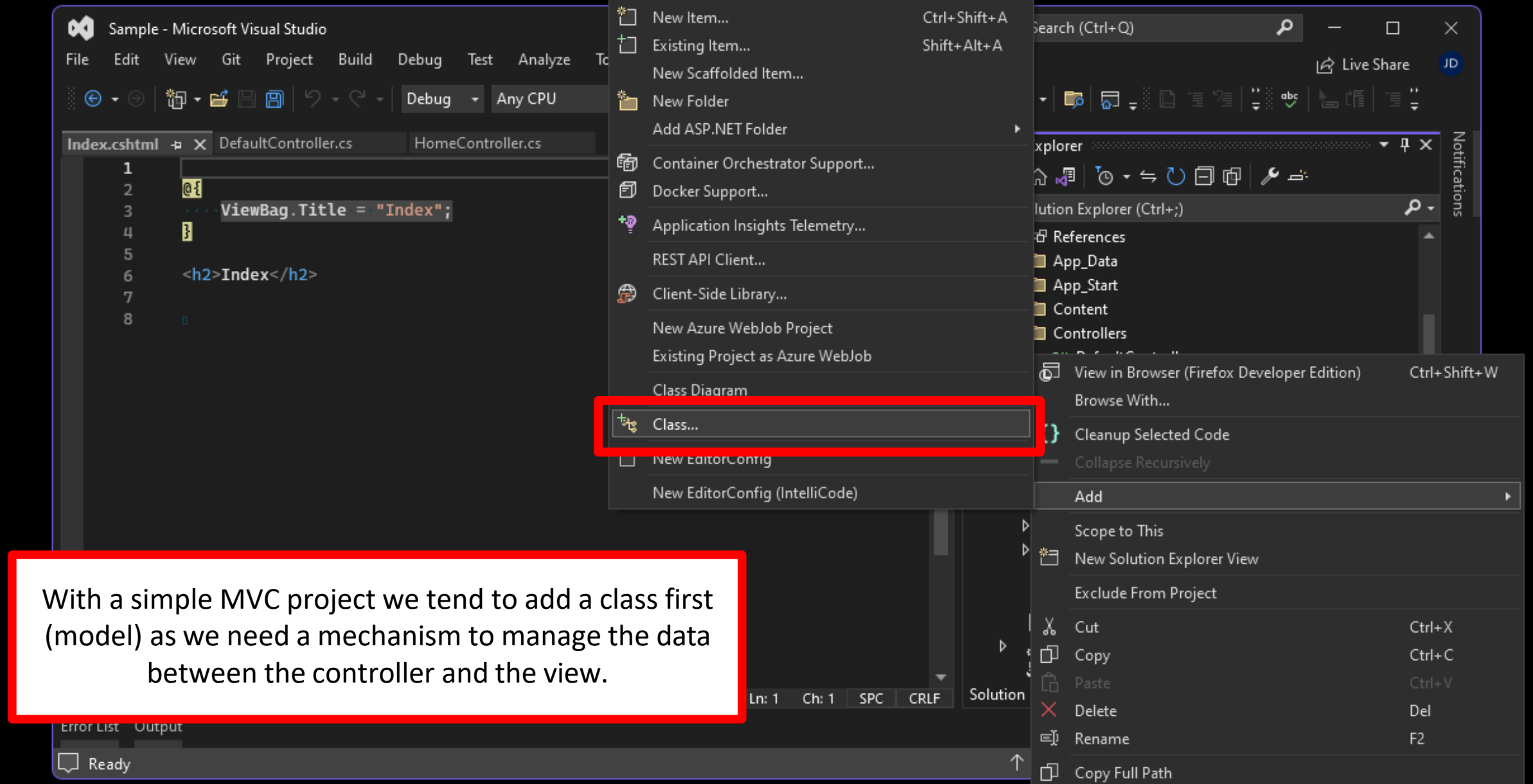
- **App_Data**: Data files, such as database files or XML.
- **App_Start**: Classes that are executed on starts up.
- **Content**: Static resources like CSS and images.
- **Controllers**: Controller classes (HTTP requests, process input, output).
- **fonts**: Font files (special or default).
- **Models**: Classes that represent the data and business logic.
- **Scripts**: JavaScript files.
- **Views**: View templates that display the user interface.
- **bin**: Contains compiled code and assemblies (hidden).
- **obj**: Temporary files generated during build (hidden).
- **Properties**: Application settings and metadata.
- **packages**: Third-party libraries and dependencies (separate folder).



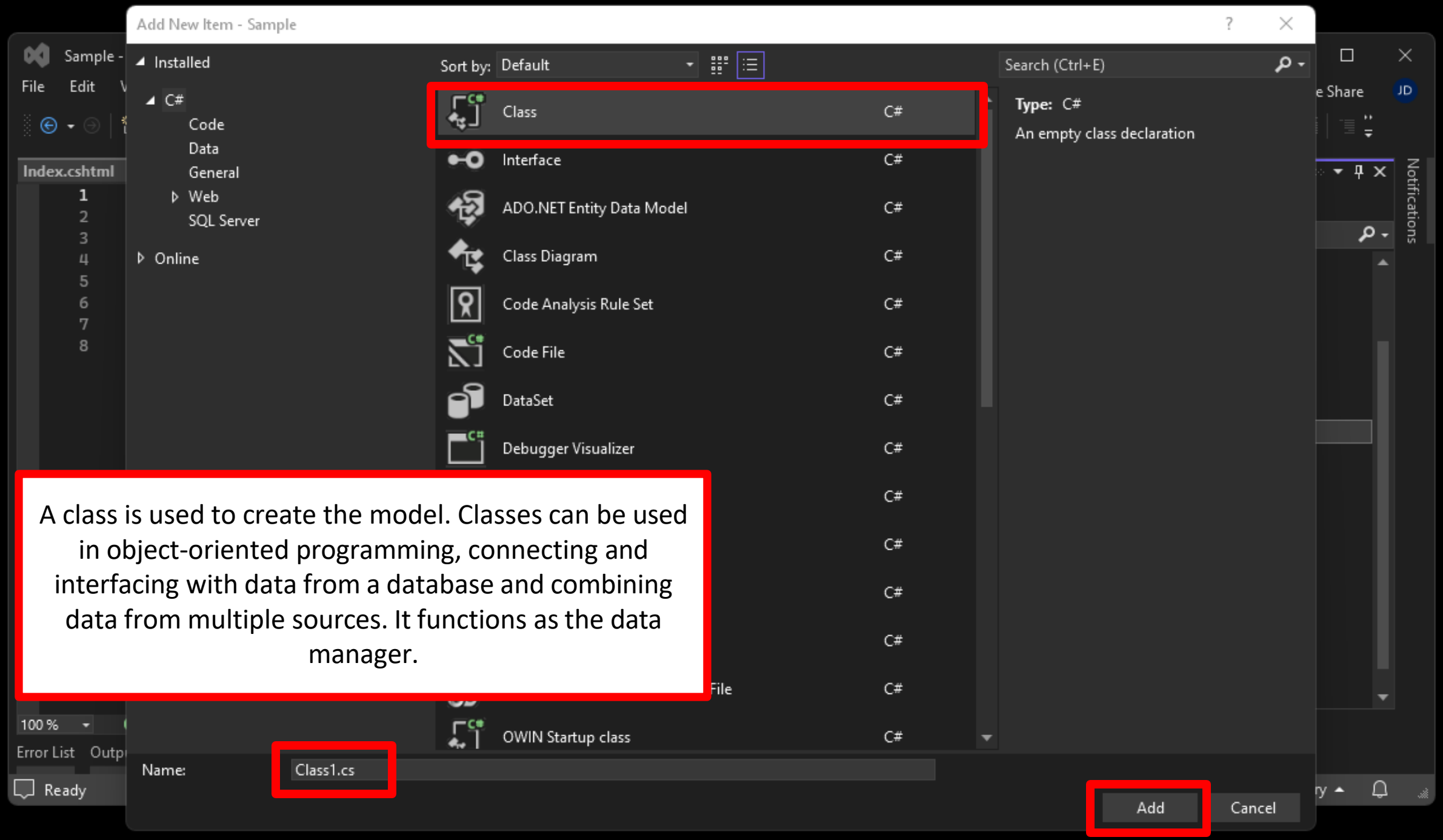
Adding new components

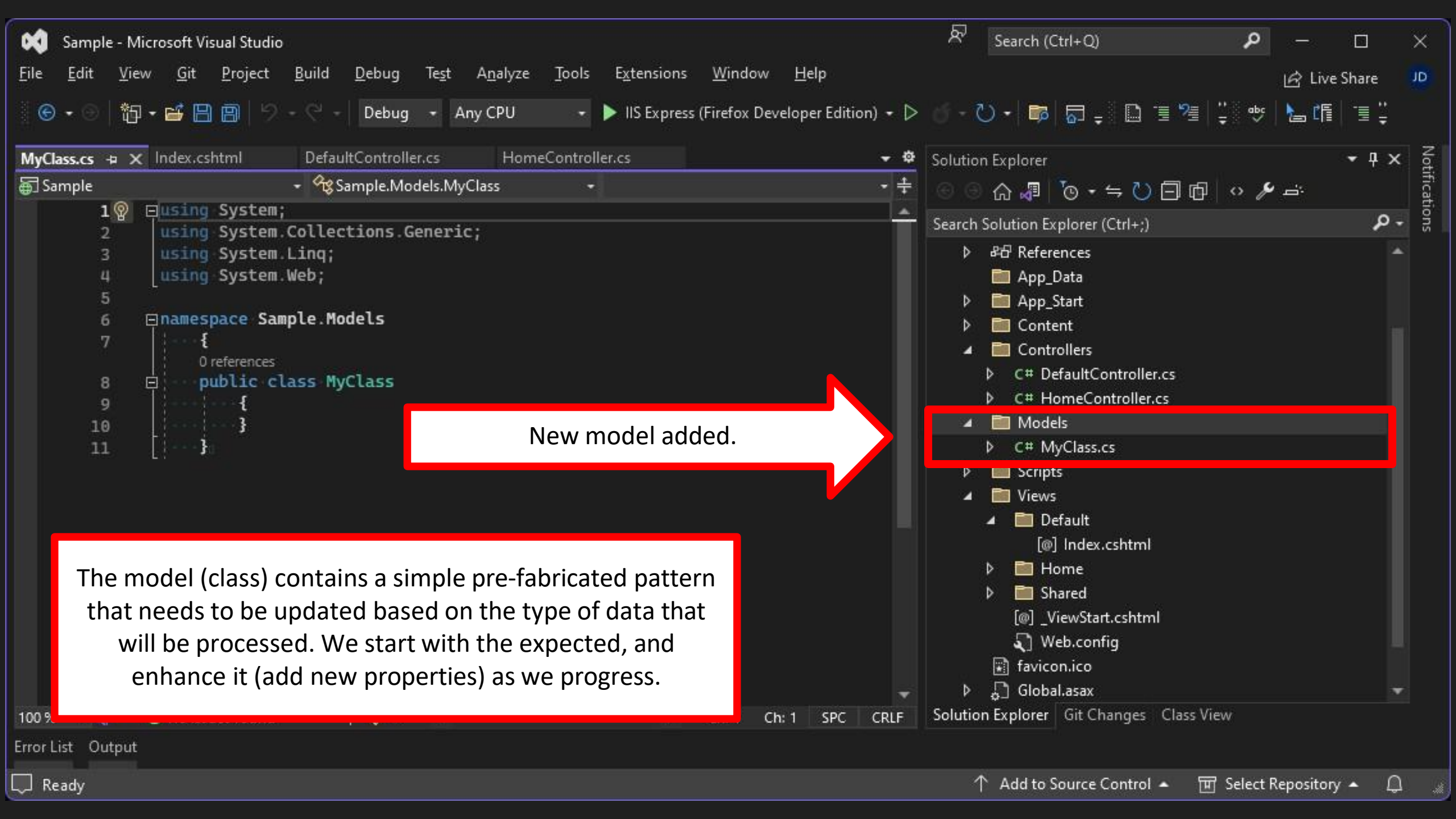
In this part we will be expanding the sample project by showing how to add additional files and where they are located.

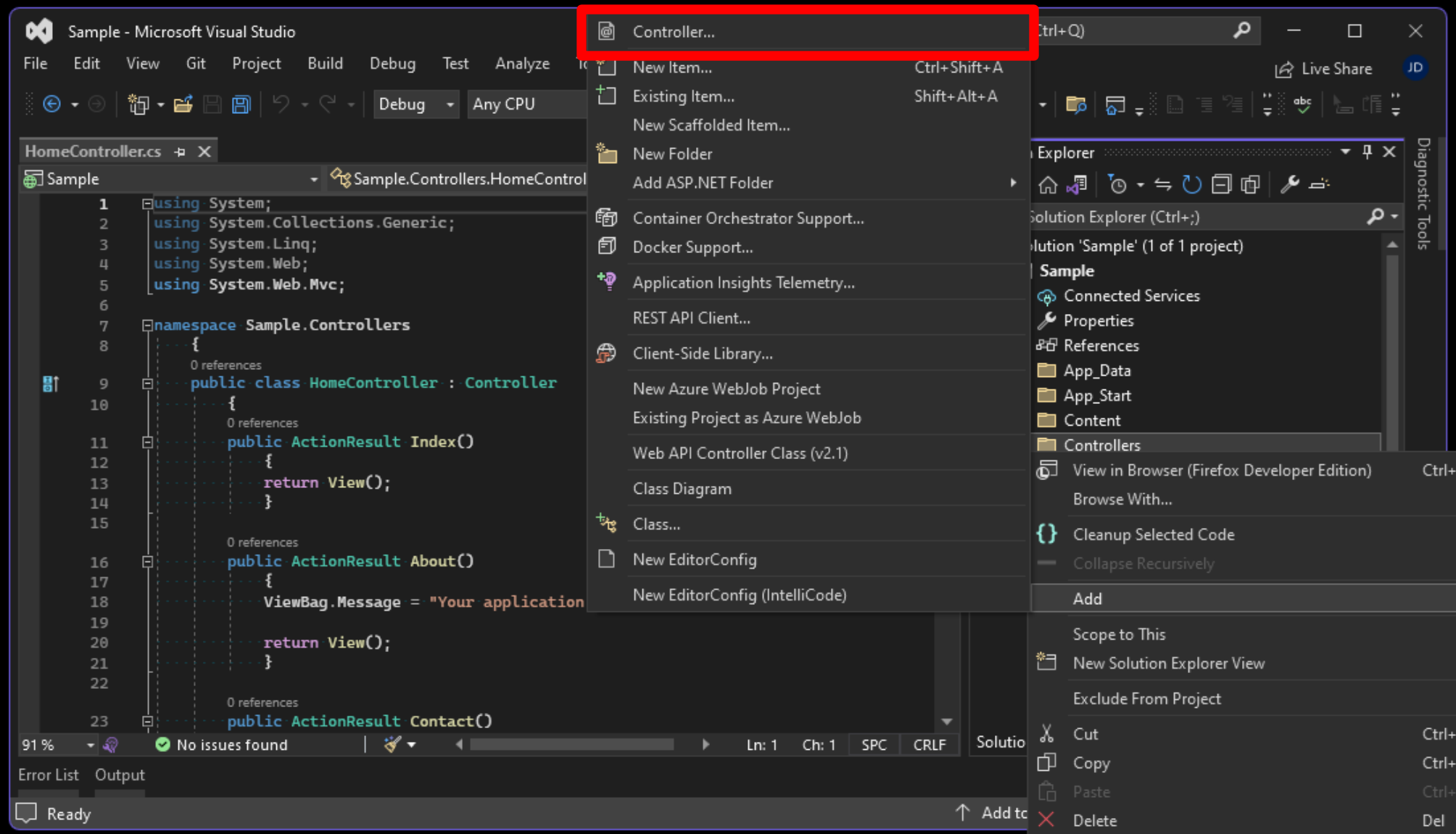

Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
JD
Debug
Any CPU
IIS Express (Firefox Developer Edition)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Sample' (1 of 1 project)
Sample
Connected Services
Properties
References
App_Data
App_Start
Content
Controllers
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
Class View
Notifications
In this section we will start with the sample pattern and add additional components. Most of the first set of activities will be simple and straightforward to make sure that you get use to the MVC environment.
Error List
Output
Ready
Add to Source Control
Select Repository



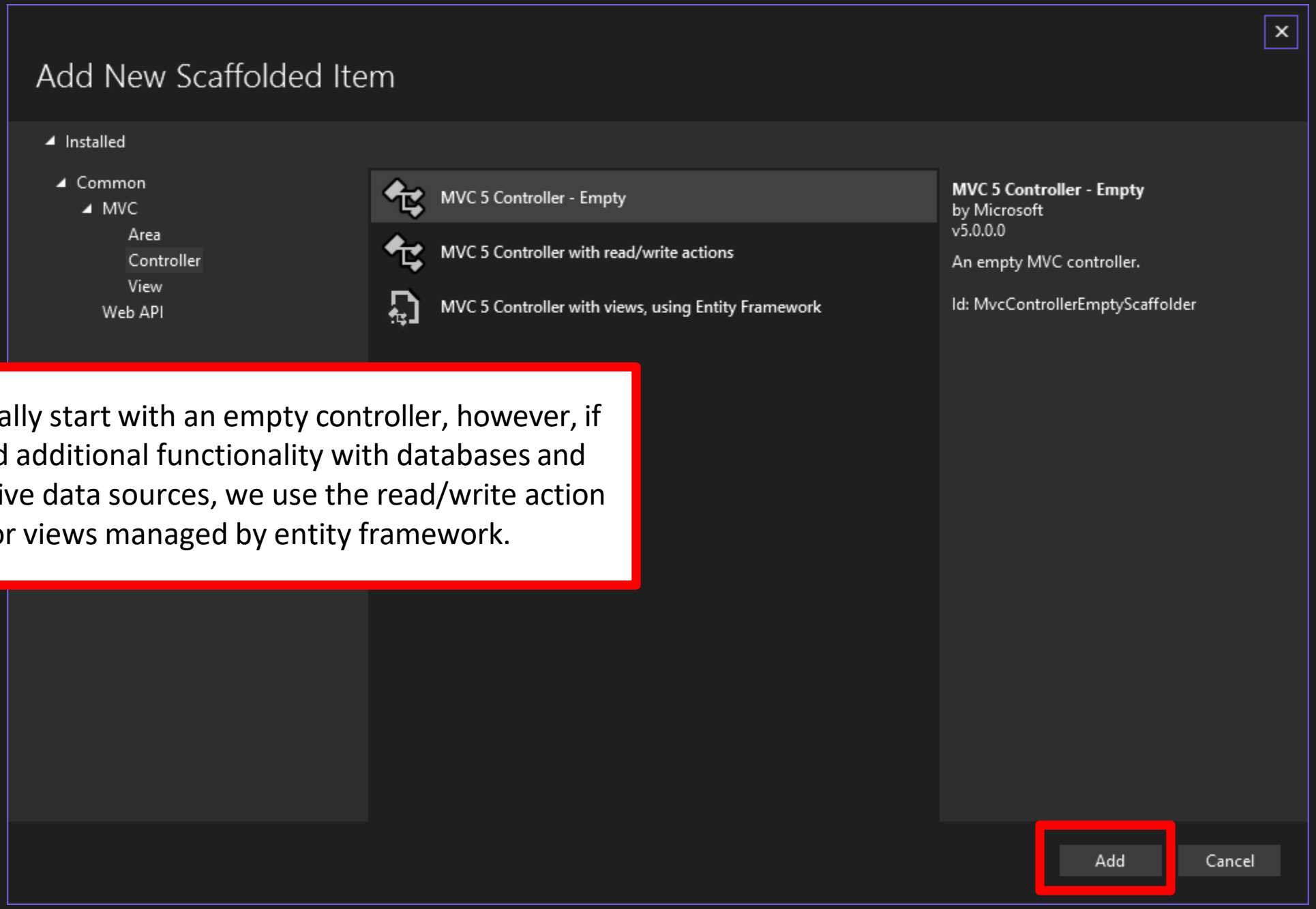
With a simple MVC project we tend to add a class first (model) as we need a mechanism to manage the data between the controller and the view.


Add New Item - Sample
Installed
C#
Code
Data
General
Web
SQL Server
Online
Sort by: Default
Search (Ctrl+E)
Class C#
Interface C#
ADO.NET Entity Data Model C#
Class Diagram C#
Code Analysis Rule Set C#
Code File C#
DataSet C#
Debugger Visualizer C#
OWIN Startup class C#
Type: C#
An empty class declaration
Name: Class1.cs
Add
Cancel
A class is used to create the model. Classes can be used in object-oriented programming, connecting and interfacing with data from a database and combining data from multiple sources. It functions as the data manager.

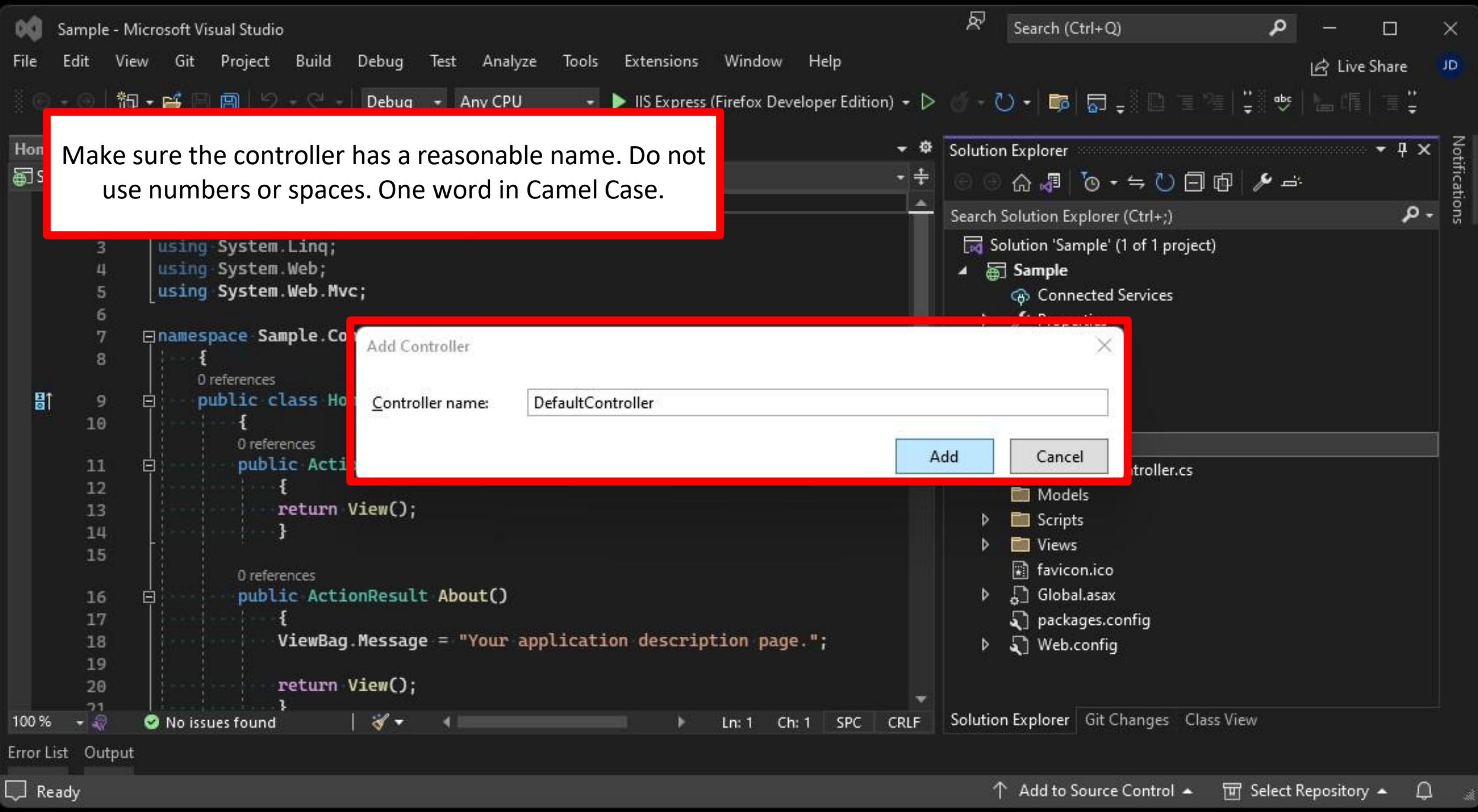

Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
MyClass.cs
Index.cshtml
DefaultController.cs
HomeController.cs
Sample
Sample.Models.MyClass
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace Sample.Models
{
0 references
public class MyClass
{
}
}
New model added.
The model (class) contains a simple pre-fabricated pattern that needs to be updated based on the type of data that will be processed. We start with the expected, and enhance it (add new properties) as we progress.
Solution Explorer
Search Solution Explorer (Ctrl+;)
References
App_Data
App_Start
Content
Controllers
DefaultController.cs
HomeController.cs
Models
MyClass.cs
Scripts
Views
Default
Index.cshtml
Home
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
Solution Explorer
Git Changes
Class View
Notifications
Ch: 1
SPC
CRLF
Error List
Output
Ready
Add to Source Control
Select Repository

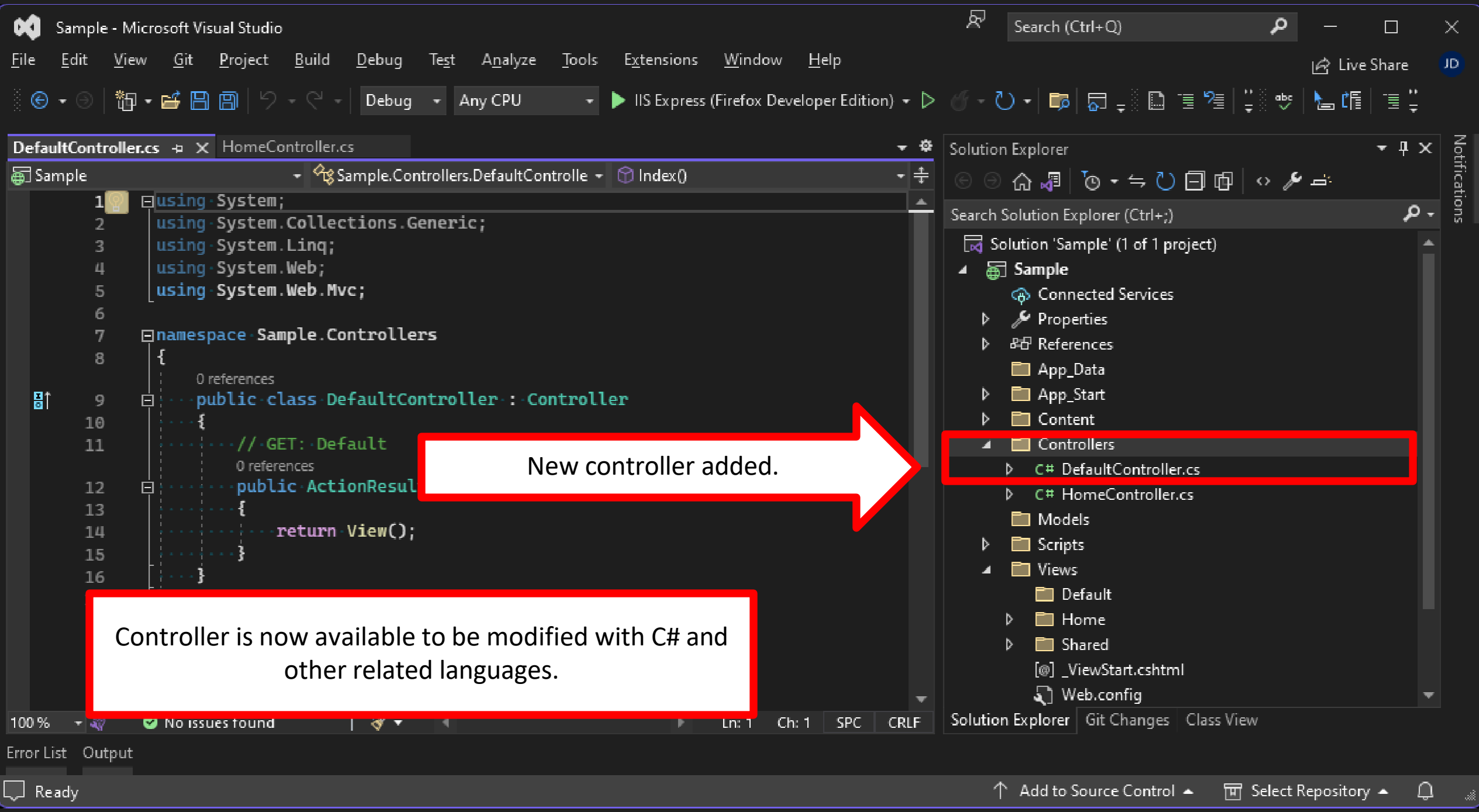


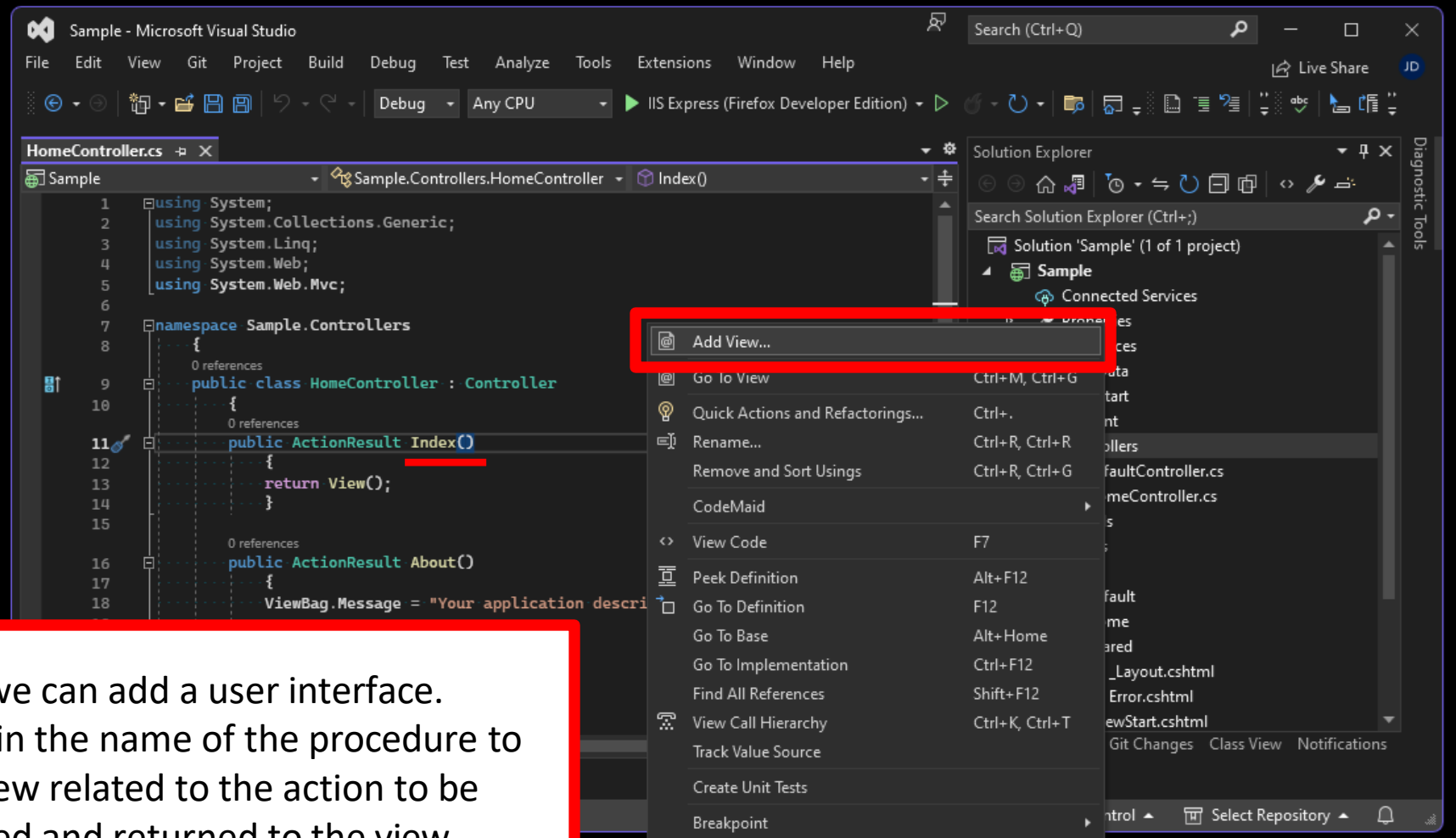
Adding a new controller to enable HTTP request from the user UI to be processed and a result to be returned.



We usually start with an empty controller, however, if we add additional functionality with databases and alternative data sources, we use the read/write action or views managed by entity framework.

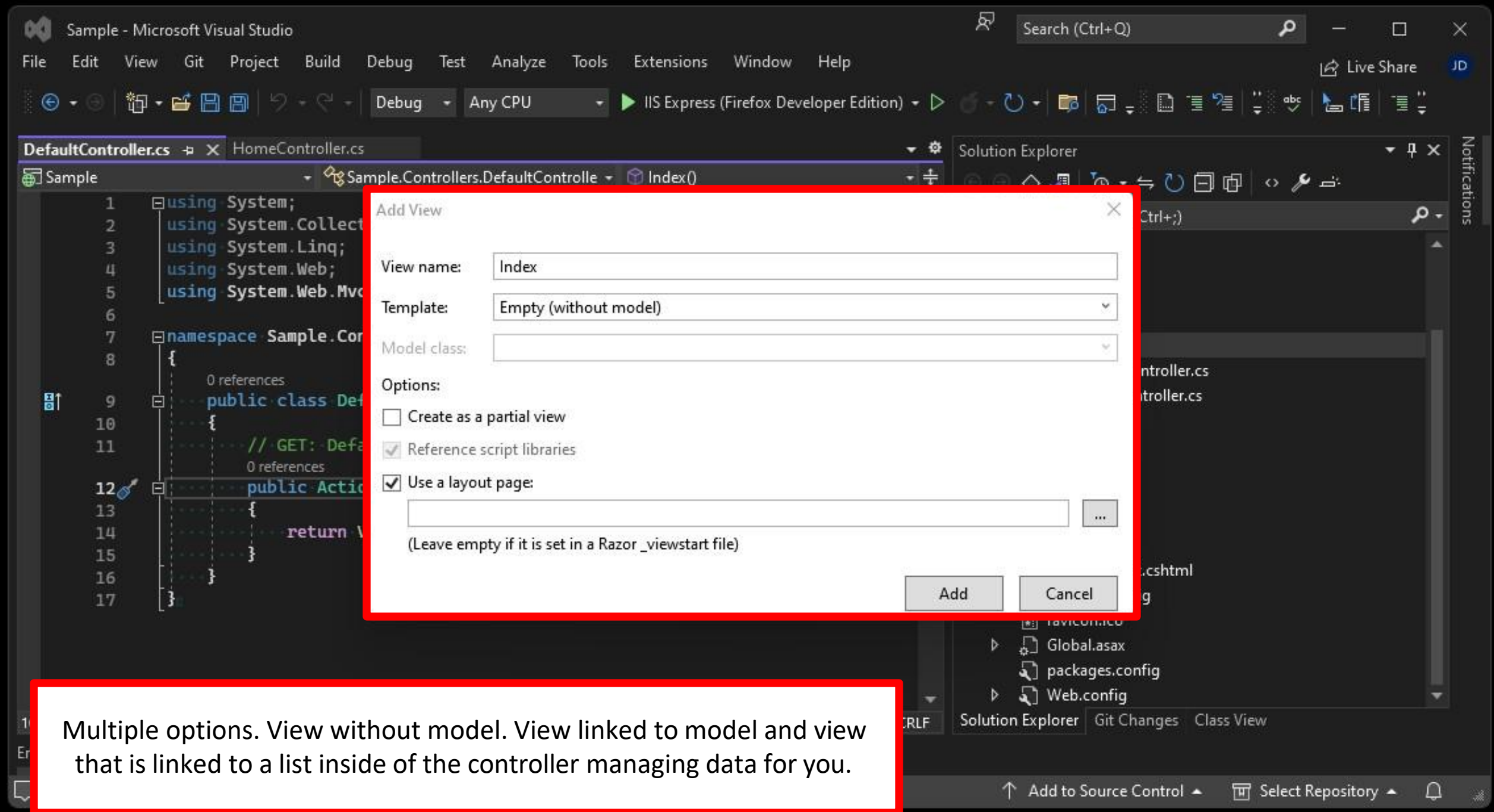

Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
Make sure the controller has a reasonable name. Do not use numbers or spaces. One word in Camel Case.
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace Sample.Co
{
0 references
public class Ho
{
0 references
public Acti
{
return View();
}
0 references
public ActionResult About()
{
ViewBag.Message = "Your application description page.";
return View();
Add Controller
Controller name:
DefaultController
Add
Cancel
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Sample' (1 of 1 project)
Sample
Connected Services
troller.cs
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
Notifications
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Solution Explorer
Git Changes
Class View
Error List
Output
Ready
Add to Source Control
Select Repository


Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
DefaultController.cs
HomeController.cs
Sample
Sample.Controllers.DefaultControlle
Index()
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace Sample.Controllers
{
0 references
public class DefaultController : Controller
{
// GET: Default
0 references
public ActionResul
{
return View();
}
}
New controller added.
Controller is now available to be modified with C# and other related languages.
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Sample' (1 of 1 project)
Sample
Connected Services
Properties
References
App_Data
App_Start
Content
Controllers
DefaultController.cs
HomeController.cs
Models
Scripts
Views
Default
Home
Shared
_ViewStart.cshtml
Web.config
Solution Explorer
Git Changes
Class View
Notifications
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Error List
Output
Ready
Add to Source Control
Select Repository

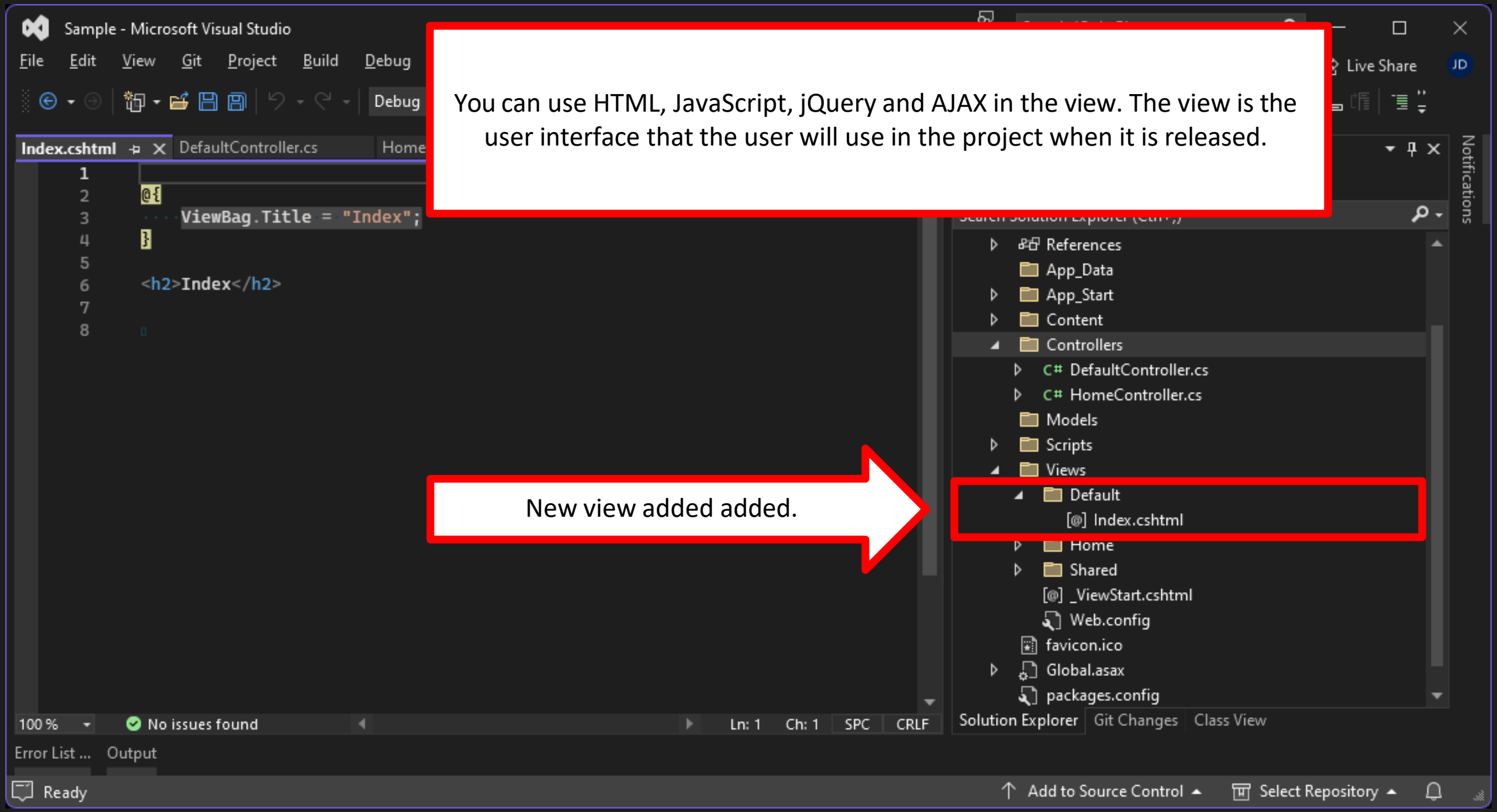


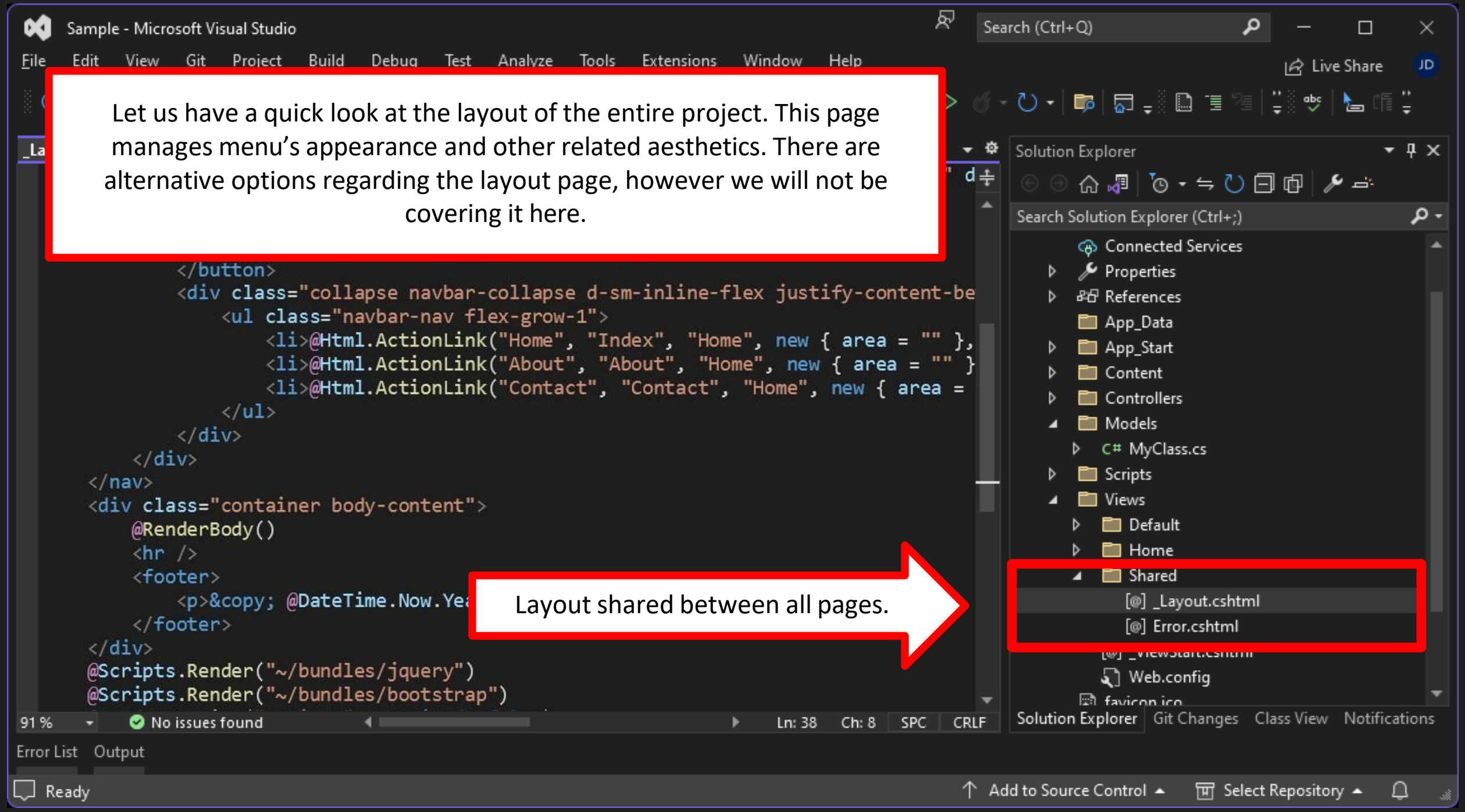
Now we can add a user interface.
Right click in the name of the procedure to add a view related to the action to be executed and returned to the view.

Think of an ActionResult as a click event procedure. The procedure receives data from the view, processes data and the returns the result back to the view.



Multiple options. View without model. View linked to model and view that is linked to a list inside of the controller managing data for you.


Sample - Microsoft Visual Studio
File Edit View Git Project Build Debug
Debug
Live Share
JD
Index.cshtml
DefaultController.cs
Home
1
2 @{
3 ViewBag.Title = "Index";
4 }
5
6 <h2>Index</h2>
7
8
You can use HTML, JavaScript, jQuery and AJAX in the view. The view is the user interface that the user will use in the project when it is released.
New view added added.
References
App_Data
App_Start
Content
Controllers
DefaultController.cs
HomeController.cs
Models
Scripts
Views
Default
Index.cshtml
Home
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Notifications
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Solution Explorer
Git Changes
Class View
Error List ...
Output
Ready
Add to Source Control
Select Repository


Sample - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
JD
Let us have a quick look at the layout of the entire project. This page manages menu's appearance and other related aesthetics. There are alternative options regarding the layout page, however we will not be covering it here.
</button>
<div class="collapse navbar-collapse d-sm-inline-flex justify-content-be
<ul class="navbar-nav flex-grow-1">
@Html.ActionLink("Home", "Index", "Home", new { area = "" },
@Html.ActionLink("About", "About", "Home", new { area = "" }
@Html.ActionLink("Contact", "Contact", "Home", new { area =

</nav>

@RenderBody()
<hr />
<footer>
<p>© @DateTime.Now.Ye
</footer>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
Layout shared between all pages.
Solution Explorer
Search Solution Explorer (Ctrl+;)
Connected Services
Properties
References
App_Data
App_Start
Content
Controllers
Models
MyClass.cs
Scripts
Views
Default
Home
Shared
_Layout.cshtml
Error.cshtml
Web.config
favicon.ico
91 %
No issues found
Ln: 38 Ch: 8 SPC CRLF
Solution Explorer Git Changes Class View Notifications
Error List Output
Ready
Add to Source Control
Select Repository

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark">
        <div class="container">
            @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"
            title="Toggle navigation" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse d-sm-inline-flex justify-content-between">
                <ul class="navbar-nav flex-grow-1">
                    <li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("About", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```

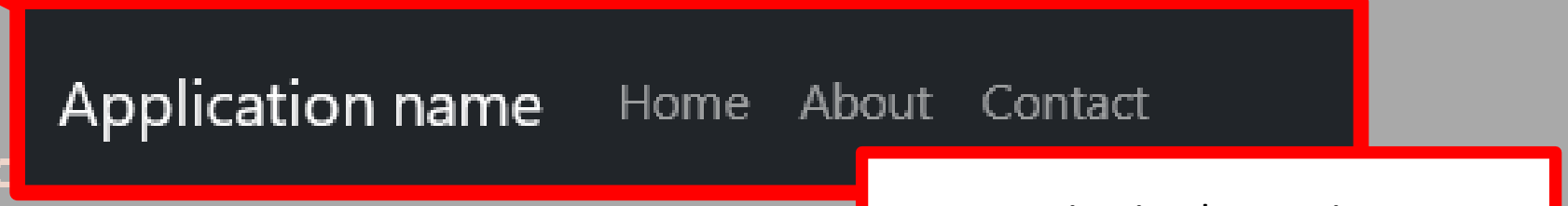
Default Razor and HTML managing the appearance of the project.

Home / project name

View linked

Controller

```
@Html.ActionLink("Application name", "Index", "Home", new { area = "" }, n
<button type="button" class="navbar-toggler" data-bs-toggle="collapse" dat
title="Toggle navigation" aria-controls="navbarSupportedContent"
        aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse d-sm-inline-flex justify-content-betw
    <ul class="navbar-nav flex-grow-1">
        <li>@Html.ActionLink("About", "About", "Home", new { area = "" },
        <li>@Html.ActionLink("Contact", "Contact", "Home", new { area = ""
    </ul>
</div>
```



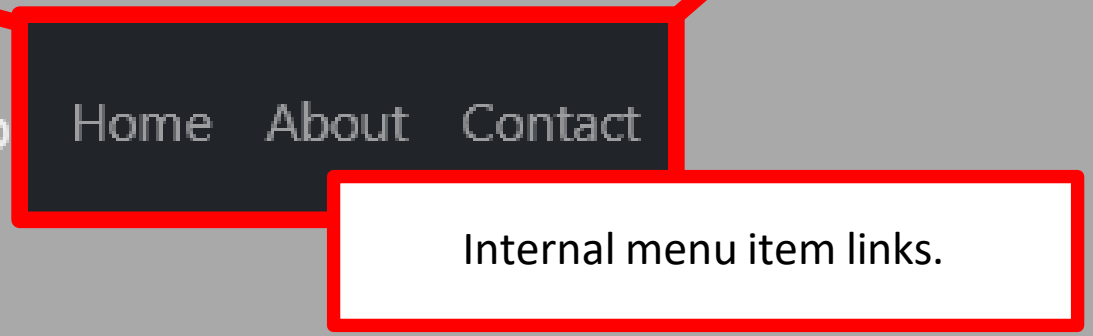
Navigation bar options.

```
class="collapse navbar-collapse d-sm-inline-flex justify-content-between">
 class="navbar-nav flex-grow-1">
 <li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @c
 <li>@Html.ActionLink("About", "About", "Home", new { area = "" }, new { @
 <li>@Html.ActionLink("Contact", "Contact", "Home", new { area = "" }, new
```

Display name

View linked

Controller



Internal menu item links.

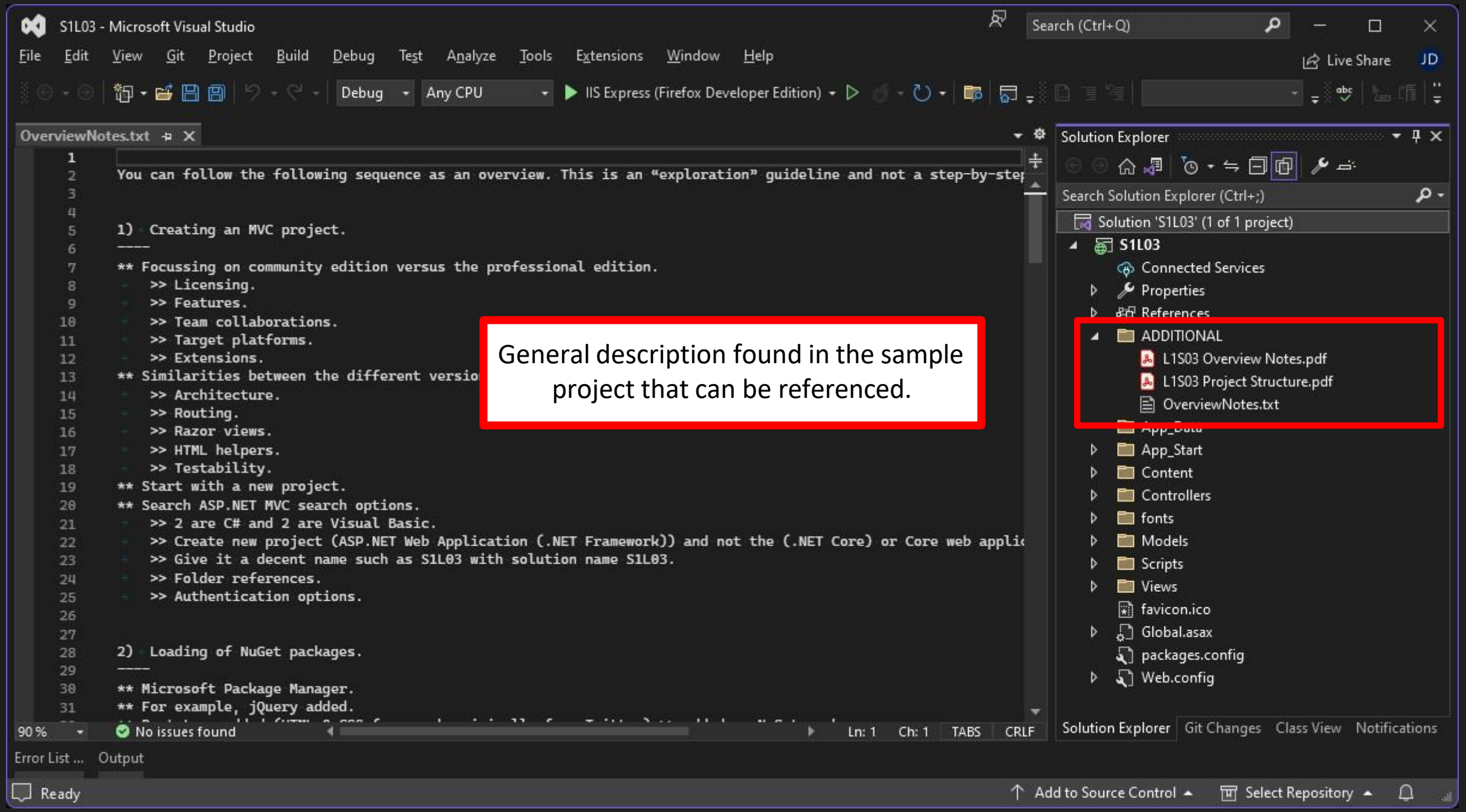
```
                <ul class="navbar-nav flex-grow-1">
                    <li>@Html.ActionLink("Home", "Index", "Home", new { area
                    <li>@Html.ActionLink("About", "About", "Home", new { area
                    <li>@Html.ActionLink("Contact", "Contact", "Home", new {
                </ul>
            </div>
        </div>
    </nav>
    <div class="container body
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```

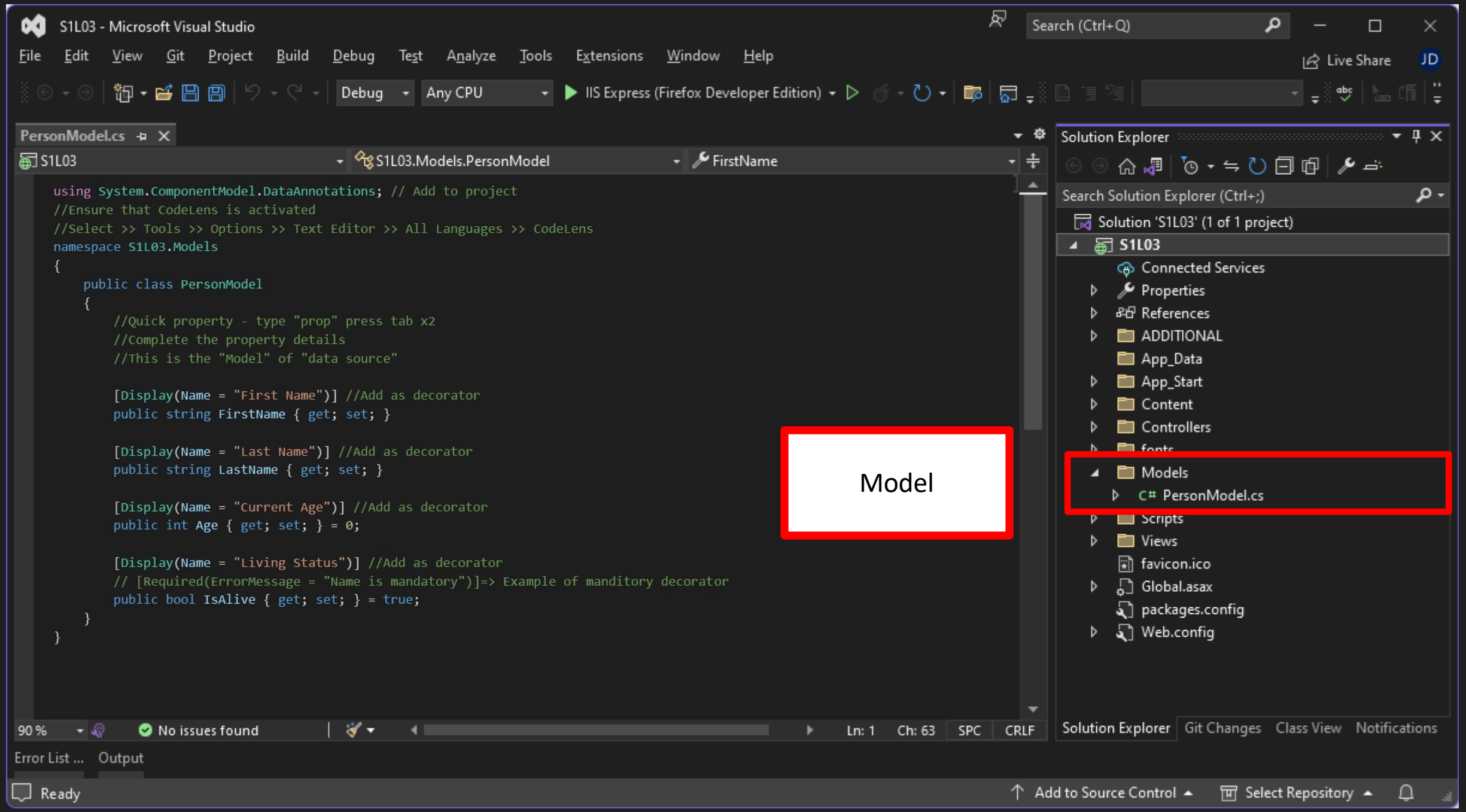
This is the command where the page is rendered. The page is inserted into the container called “Render Section”

This is also where global scripts as stipulated.

Simple MVC project

In this example, a specific MVC project named S1L03 has been developed and the indicated process was used to set the fundamental details. So we are only going to focus on the application code.


S1L03 - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
JD
Debug
Any CPU
IIS Express (Firefox Developer Edition)
OverviewNotes.txt
You can follow the following sequence as an overview. This is an "exploration" guideline and not a step-by-step
1) Creating an MVC project.
** Focussing on community edition versus the professional edition.
>> Licensing.
>> Features.
>> Team collaborations.
>> Target platforms.
>> Extensions.
** Similarities between the different versio
>> Architecture.
>> Routing.
>> Razor views.
>> HTML helpers.
>> Testability.
** Start with a new project.
** Search ASP.NET MVC search options.
>> 2 are C# and 2 are Visual Basic.
>> Create new project (ASP.NET Web Application (.NET Framework)) and not the (.NET Core) or Core web applic
>> Give it a decent name such as S1L03 with solution name S1L03.
>> Folder references.
>> Authentication options.
2) Loading of NuGet packages.
** Microsoft Package Manager.
** For example, jQuery added.
General description found in the sample project that can be referenced.
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'S1L03' (1 of 1 project)
S1L03
Connected Services
Properties
References
ADDITIONAL
L1S03 Overview Notes.pdf
L1S03 Project Structure.pdf
OverviewNotes.txt
App_Start
Content
Controllers
fonts
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
Class View
Notifications
90 %
No issues found
Ln: 1
Ch: 1
TABS
CRLF
Error List ...
Output
Ready
Add to Source Control
Select Repository

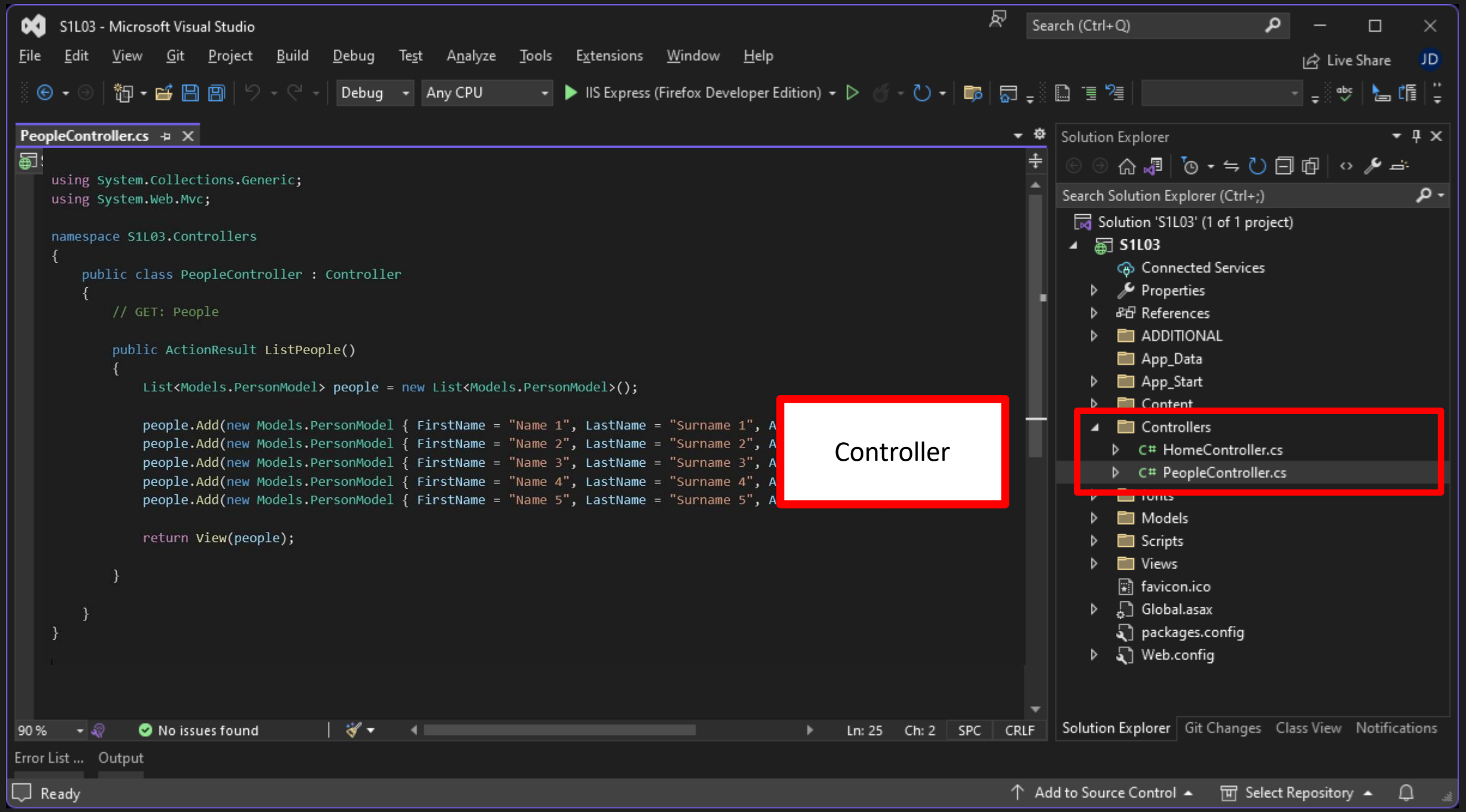

S1L03 - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
PersonModel.cs
S1L03
S1L03.Models.PersonModel
FirstName
using System.ComponentModel.DataAnnotations; // Add to project
//Ensure that CodeLens is activated
//Select >> Tools >> Options >> Text Editor >> All Languages >> CodeLens
namespace S1L03.Models
{
public class PersonModel
{
//Quick property - type "prop" press tab x2
//Complete the property details
//This is the "Model" of "data source"
[Display(Name = "First Name")] //Add as decorator
public string FirstName { get; set; }
[Display(Name = "Last Name")] //Add as decorator
public string LastName { get; set; }
[Display(Name = "Current Age")] //Add as decorator
public int Age { get; set; } = 0;
[Display(Name = "Living Status")] //Add as decorator
// [Required(ErrorMessage = "Name is mandatory")]=> Example of manditory decorator
public bool IsAlive { get; set; } = true;
}
}
Model
90 %
No issues found
Ln: 1
Ch: 63
SPC
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'S1L03' (1 of 1 project)
S1L03
Connected Services
Properties
References
ADDITIONAL
App_Data
App_Start
Content
Controllers
Models
PersonModel.cs
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
Class View
Notifications
Error List ...
Output
Ready
Add to Source Control
Select Repository

```
using System.ComponentModel.DataAnnotations; // Add to project
//Ensure that CodeLens is activated
//Select >> Tools >> Options >> Text
namespace S1L03.Models
{
    public class PersonModel
    {
        //Quick property - type "prop" press tab x2
        //Complete the property deta
        //This is the "Model" of "da

        [Display(Name = "First Name")] //Add as decorator
        public string FirstName { get; set; }

        [Display(Name = "Last Name")] //Add as decorator
        public string LastName { get; set; }

        [Display(Name = "Current Age")] //Add as decorator
        public int Age { get; set; } = 0;

        [Display(Name = "Living Status")] //Add as decorator
        // [Required(ErrorMessage = "Name is mandatory")]=> Example of manditory decorator
        public bool IsAlive { get; set; } = true;
    }
}
```

Decorator = Display | Control

```
[Display(Name = "First Name")]
public string FirstName { get; set; }
```

Access	Data type	Variable Name	Obtain | Prepare


S1L03 - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
PeopleController.cs
using System.Collections.Generic;
using System.Web.Mvc;
namespace S1L03.Controllers
{
public class PeopleController : Controller
{
// GET: People
public ActionResult ListPeople()
{
List<Models.PersonModel> people = new List<Models.PersonModel>();
people.Add(new Models.PersonModel { FirstName = "Name 1", LastName = "Surname 1", A
people.Add(new Models.PersonModel { FirstName = "Name 2", LastName = "Surname 2", A
people.Add(new Models.PersonModel { FirstName = "Name 3", LastName = "Surname 3", A
people.Add(new Models.PersonModel { FirstName = "Name 4", LastName = "Surname 4", A
people.Add(new Models.PersonModel { FirstName = "Name 5", LastName = "Surname 5", A
return View(people);
}
}
}
Controller
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'S1L03' (1 of 1 project)
S1L03
Connected Services
Properties
References
ADDITIONAL
App_Data
App_Start
Content
Controllers
HomeController.cs
PeopleController.cs
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Web.config
90 %
No issues found
Ln: 25 Ch: 2 SPC CRLF
Solution Explorer Git Changes Class View Notifications
Error List ... Output
Ready
Add to Source Control
Select Repository

```
using System.Collections.Generic;
using System.Web.Mvc;

namespace S1L03.Controllers
{
    {
        {
            List<Models.PersonModel> people = new List<Models.PersonModel>();

            people.Add(new Models.PersonModel { FirstName = "Name 1", LastName = "Surname 1", Age = 1, IsAlive = true });
            people.Add(new Models.PersonModel { FirstName = "Name 2", LastName = "Surname 2", Age = 2, IsAlive = false });
            people.Add(new Models.PersonModel { FirstName = "Name 3", LastName = "Surname 3", Age = 3, IsAlive = true });
            people.Add(new Models.PersonModel { FirstName = "Name 4", LastName = "Surname 4", Age = 4, IsAlive = false });
            people.Add(new Models.PersonModel { FirstName = "Name 5", LastName = "Surname 5", Age = 5, IsAlive = true });
```

Take data in the list, create new list and copy data to the new list as new data is added.

Models = Folder

PersonModel = Class

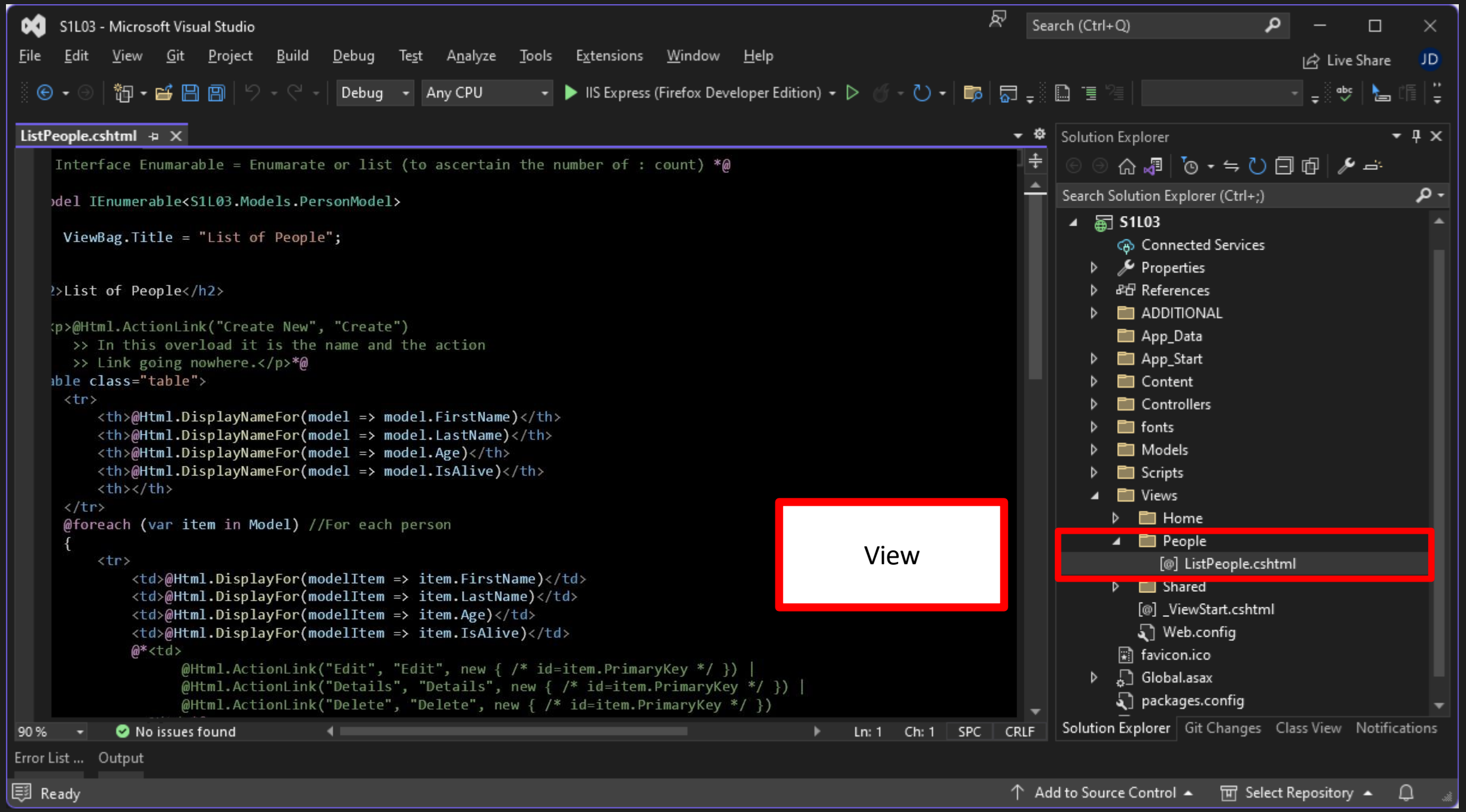
people = variable

New data from model

Add data to variable

Add new data to model list

Values being added according to the sequence in the model


S1L03 - Microsoft Visual Studio
Search (Ctrl+Q)
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Live Share
Debug
Any CPU
IIS Express (Firefox Developer Edition)
ListPeople.cshtml
Interface Enumarable = Enumarate or list (to ascertain the number of : count) *@
odel IEnumerable<S1L03.Models.PersonModel>
ViewBag.Title = "List of People";
>List of People</h2>
p>@Html.ActionLink("Create New", "Create")
>> In this overload it is the name and the action
>> Link going nowhere.</p>*@
able class="table">
<tr>
<th>@Html.DisplayNameFor(model => model.FirstName)</th>
<th>@Html.DisplayNameFor(model => model.LastName)</th>
<th>@Html.DisplayNameFor(model => model.Age)</th>
<th>@Html.DisplayNameFor(model => model.IsAlive)</th>
<th></th>
</tr>
@foreach (var item in Model) //For each person
{
<tr>
<td>@Html.DisplayFor(modelItem => item.FirstName)</td>
<td>@Html.DisplayFor(modelItem => item.LastName)</td>
<td>@Html.DisplayFor(modelItem => item.Age)</td>
<td>@Html.DisplayFor(modelItem => item.IsAlive)</td>
@*<td>
@Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ }) |
@Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey */ }) |
@Html.ActionLink("Delete", "Delete", new { /* id=item.PrimaryKey */ })
View
90 %
No issues found
Ln: 1 Ch: 1 SPC CRLF
Error List ... Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
S1L03
Connected Services
Properties
References
ADDITIONAL
App_Data
App_Start
Content
Controllers
fonts
Models
Scripts
Views
Home
People
ListPeople.cshtml
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Solution Explorer Git Changes Class View Notifications
Ready
Add to Source Control
Select Repository

```
@* Interface Enumarable = Enumarate or list (to ascertain the number of : count) *@

@model IEnumerable<S1L03.Models.PersonModel>
@{
    ViewBag.Title = "List of People";
 }
```

Take the decorator (if there is one) from the model property and display it as HTML in a browser

Generate HTML	and display decorator from	the model	Where the model holds…

```
    <tr>
        <th>@Html.DisplayNameFor(model => model.FirstName)</th>
        <th>@Html.DisplayNameFor(model => model.LastName)</th>
        <th>@Html.DisplayNameFor(model => model.Age)</th>
        <th>@Html.DisplayNameFor(model => model.IsAlive)</th>
        <th></th>
    </tr>
    @foreach (var item in Model) //For each person
    {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.FirstName)</td>
            <td>@Html.DisplayFor(modelItem => item.LastName)</td>
            <td>@Html.DisplayFor(modelItem => item.Age)</td>
            <td>@Html.DisplayFor(modelItem => item.IsAlive)</td>
            @*<td>
```

Generate HTML	and display data from	the model	Where the model holds…

```
            </td>*@
```

Take the data that is stored in the model property and display it as HTML in a browser

Loop through the data in the model (list) and add a new row in the table as the data increments.

File Edit View History Bookmarks Tools Help

List of People - My ASP.NET Ap

Take the decorator (if there is one) from the model property and display it as HTML in a browser

Generate HTML	and display decorator from	the model	Where the model holds…

Result

First Name	Last Name	Current Age	Living Status
Name 1	Surname 1	1	☑
Name 2	Surname 2	2	☐
Name 3	Surname 3	3	☑
Name 4	Surname 4	4	☐
Name 5	Surname 5	5	☑

© 2023 - My ASP.NET Application

Generate HTML	and display data from	the model	Where the model holds…

Take the data that is stored in the model property and display it as HTML in a browser

Loop through the data in the model (list) and add a new row in the table as the data increments.

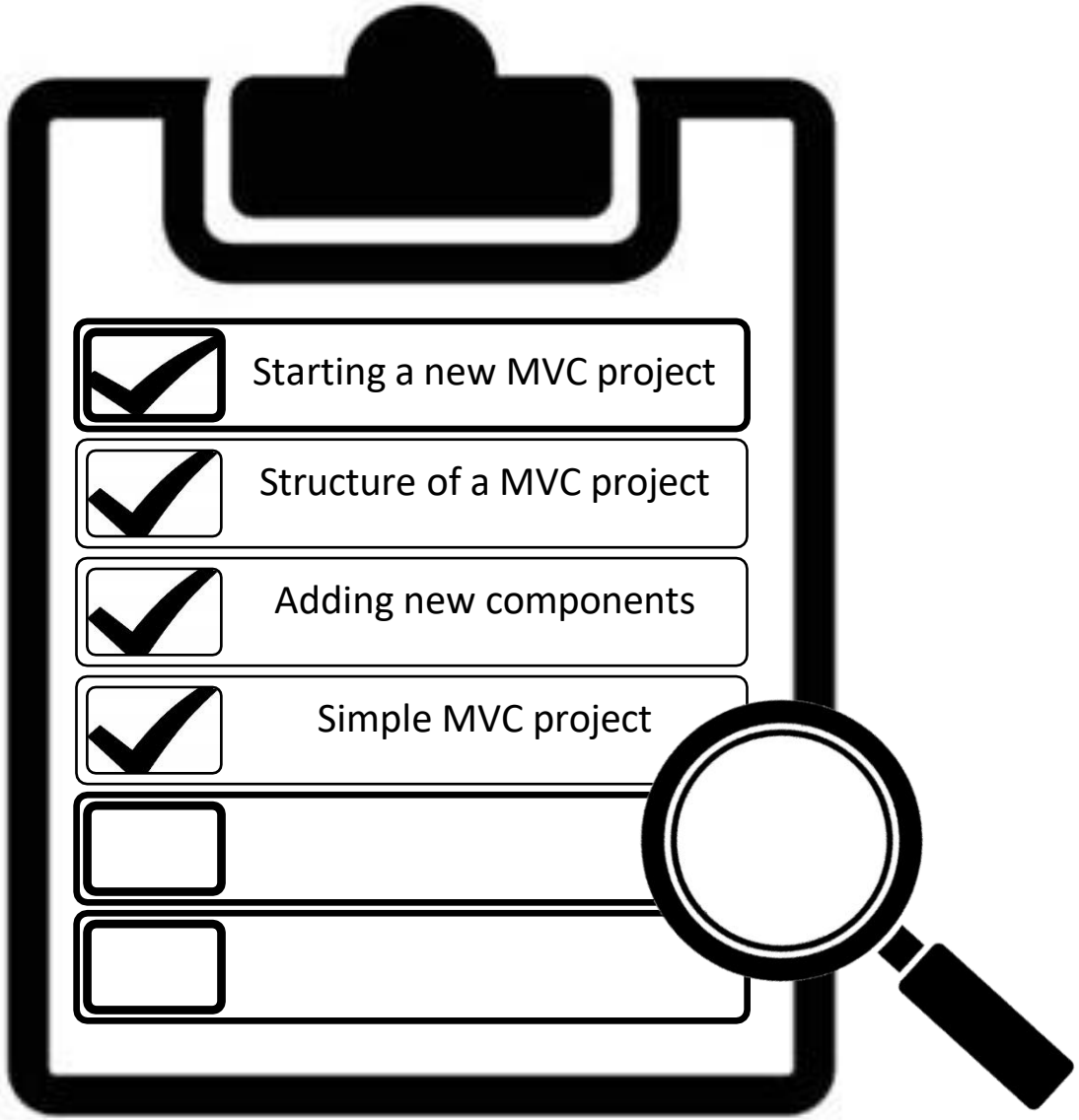
Practical overview

This is a simple brief overview of the practical work that will be completed in the practical sessions this week.

Main tasks

- Create a new MVC project named after you (myStudyGroup).
- Create a model that relates to the table indicated below.
- Create a controller that relates to the table indicated below.
- Create a view related to the table indicated below.
- Add data to the controller that relates to your INF272 study support group.
- Make sure that you have at least 5 names in the list.
- Link the newly create view to the Layout.
- Load the final program to your GitHub account.
- Upload the GitHub hyperlink to the ClickUP 2025-03-03 (before 23:59) – This is a hard cut-off.
- There are no extensions in INF272.

STUDENT NUMBER	NAME	SURNAME	EMAIL ADDRESS
Your student number	Your name	Your surname	Your email address
Member 1	….etc		
Member 2		…etc	
Member 3			…etc
Member 3			


Starting a new MVC project
Structure of a MVC project
Adding new components
Simple MVC project


So what did we cover today?

ANY Questions?