Strongly Typed View in ASP.NET Core MVC

.

Why do we need Strongly Typed View in ASP.NET Core MVC?

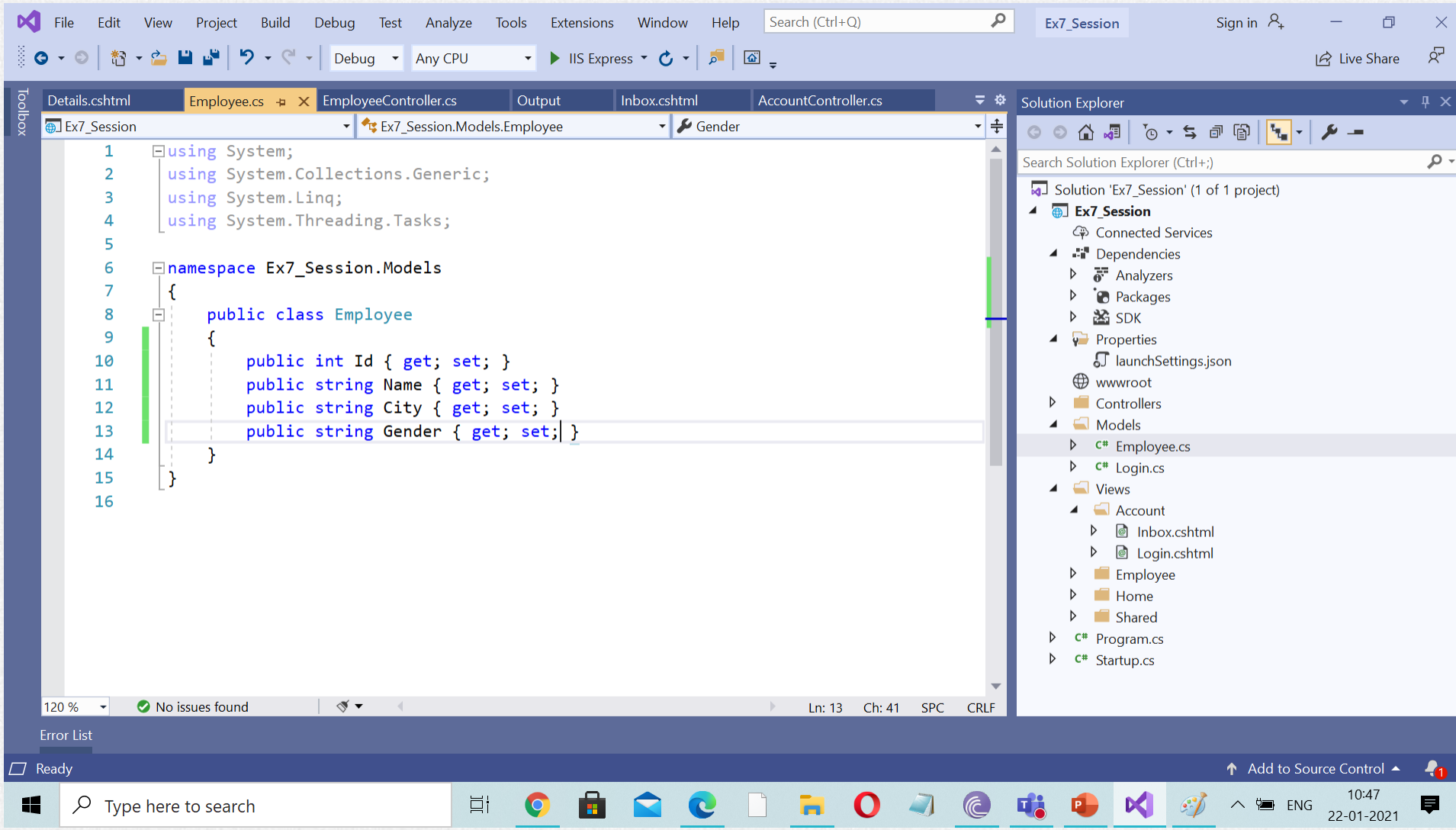
- As we already discussed we can pass the model data to a view using many different ways such as ViewBag, ViewData, strongly typed model object, etc.
- When we passed the model data to a View using ViewBag or ViewData, then the view becomes a loosely typed view.
- In a loosely typed view, we will not get any intelligence as well as the compile-time error.
- With a strongly typed view, we will get both intelligence support as well as the compile-time error.

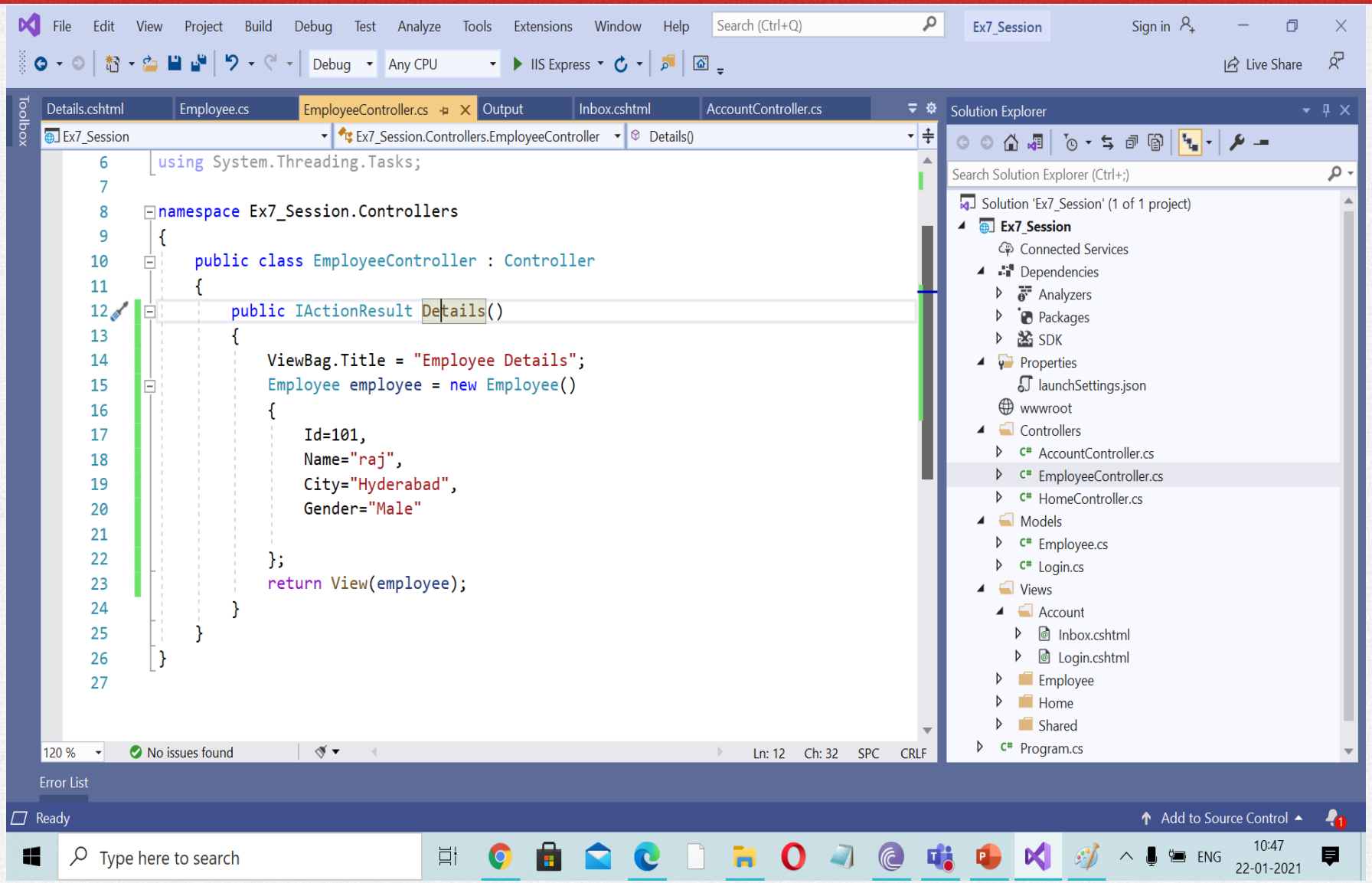
Implementing Strongly Typed View in ASP.NET Core MVC

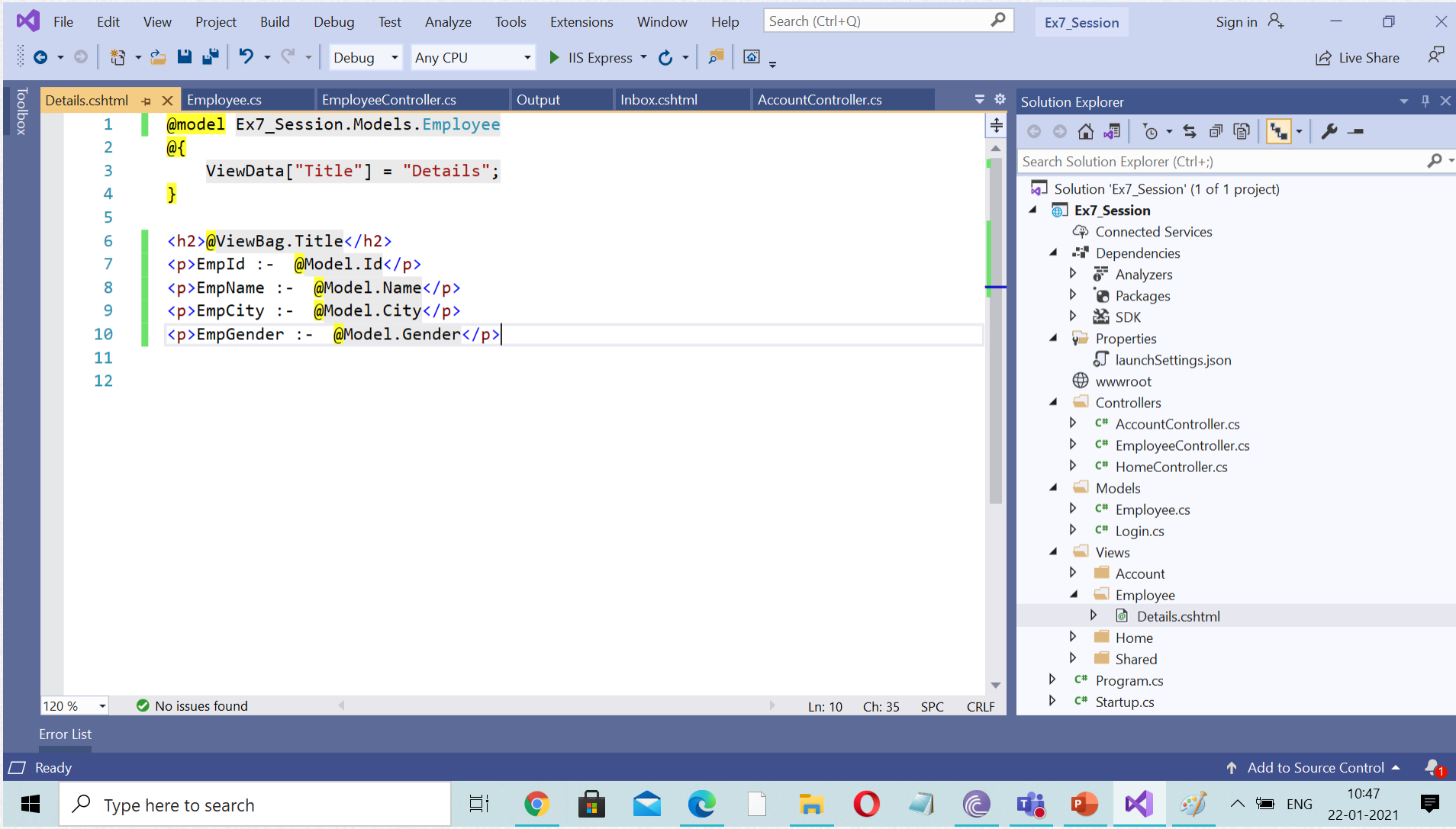
- In order to create a strongly-typed view, from the action method of the controller, we need to pass the model object as a parameter to the View() extension method.
- The Controller base class provides us the following two overloaded versions of View() extension method which we can use to pass the model object from the controller action method to a view.

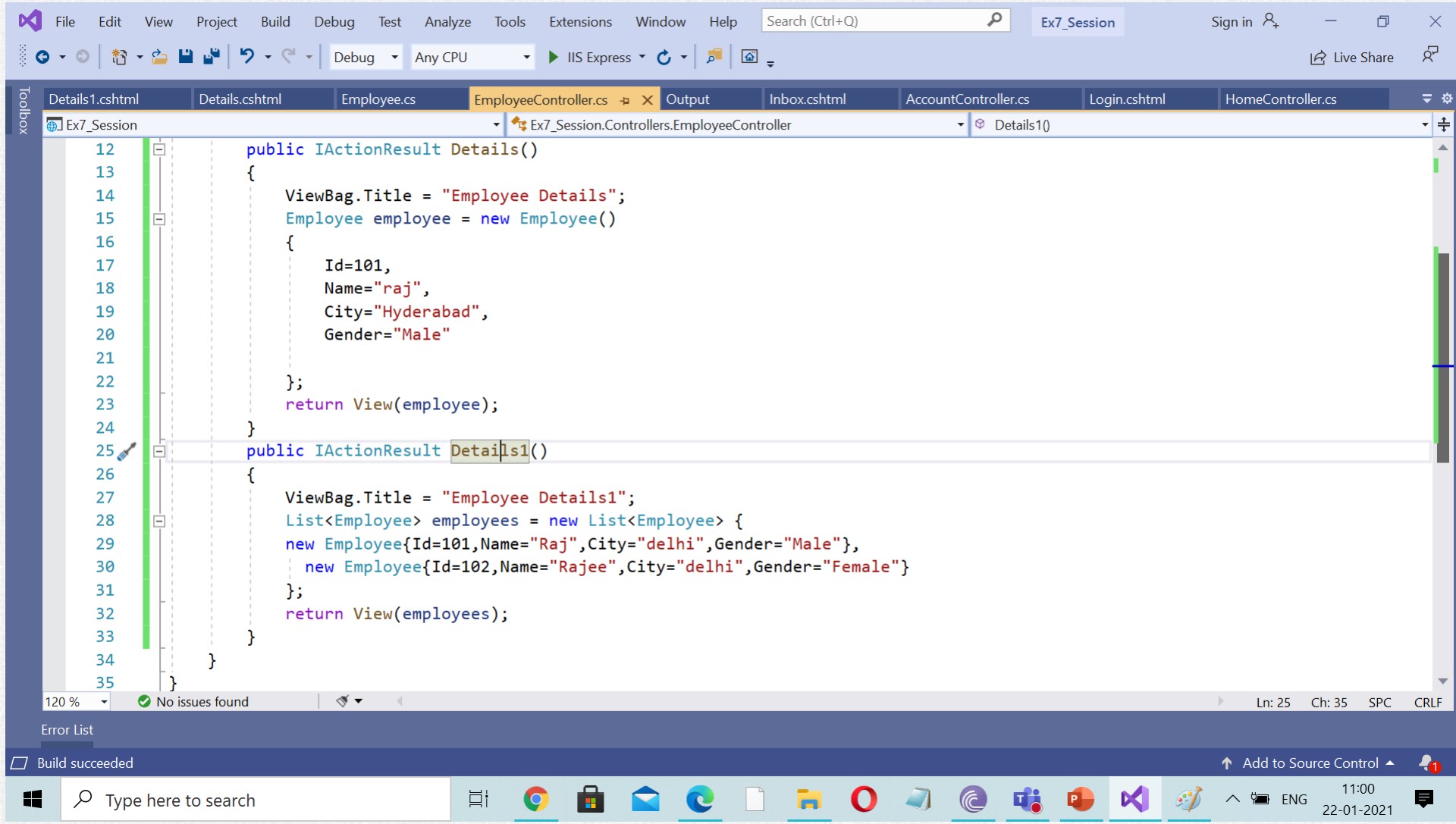
```
public virtual ViewResult View(string viewName, object model);
public virtual ViewResult View(object model);
```

- Here we are going to use the overloaded version which takes only the model object as an input parameter.


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Ex7_Session
Sign in
Debug
Any CPU
IIS Express
Live Share
Toolbox
Details.cshtml
Employee.cs
EmployeeController.cs
Output
Inbox.cshtml
AccountController.cs
Ex7_Session
Ex7_Session.Models.Employee
Gender
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace Ex7_Session.Models
7 {
8 public class Employee
9 {
10 public int Id { get; set; }
11 public string Name { get; set; }
12 public string City { get; set; }
13 public string Gender { get; set; }
14 }
15 }
16
120 %
No issues found
Ln: 13
Ch: 41
SPC
CRLF
Error List
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Ex7_Session' (1 of 1 project)
Ex7_Session
Connected Services
Dependencies
Analyzers
Packages
SDK
Properties
launchSettings.json
wwwroot
Controllers
Models
Employee.cs
Login.cs
Views
Account
Inbox.cshtml
Login.cshtml
Employee
Home
Shared
Program.cs
Startup.cs
Ready
Add to Source Control
Type here to search
ENG
10:47
22-01-2021


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Ex7_Session
Sign in
Debug
Any CPU
IIS Express
Live Share
Details.cshtml
Employee.cs
EmployeeController.cs
Output
Inbox.cshtml
AccountController.cs
Ex7_Session
Ex7_Session.Controllers.EmployeeController
Details()
using System.Threading.Tasks;
namespace Ex7_Session.Controllers
{
public class EmployeeController : Controller
{
public IActionResult Details()
{
ViewBag.Title = "Employee Details";
Employee employee = new Employee()
{
Id=101,
Name="raj",
City="Hyderabad",
Gender="Male"
};
return View(employee);
}
}
}
120 %
No issues found
Ln: 12
Ch: 32
SPC
CRLF
Error List
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Ex7_Session' (1 of 1 project)
Ex7_Session
Connected Services
Dependencies
Analyzers
Packages
SDK
Properties
launchSettings.json
wwwroot
Controllers
AccountController.cs
EmployeeController.cs
HomeController.cs
Models
Employee.cs
Login.cs
Views
Account
Inbox.cshtml
Login.cshtml
Employee
Home
Shared
Program.cs
Ready
Add to Source Control
Type here to search
ENG
10:47
22-01-2021


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Ex7_Session
Sign in
Debug
Any CPU
IIS Express
Live Share
Toolbox
Details.cshtml
Employee.cs
EmployeeController.cs
Output
Inbox.cshtml
AccountController.cs
1 @model Ex7_Session.Models.Employee
2 @{
3 ViewData["Title"] = "Details";
4 }
5
6 <h2>@ViewBag.Title</h2>
7 EmpId :- @Model.Id
8 EmpName :- @Model.Name
9 EmpCity :- @Model.City
10 EmpGender :- @Model.Gender
11
12
120 %
No issues found
Ln: 10 Ch: 35 SPC CRLF
Error List
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Ex7_Session' (1 of 1 project)
Ex7_Session
Connected Services
Dependencies
Analyzers
Packages
SDK
Properties
launchSettings.json
wwwroot
Controllers
AccountController.cs
EmployeeController.cs
HomeController.cs
Models
Employee.cs
Login.cs
Views
Account
Employee
Details.cshtml
Home
Shared
Program.cs
Startup.cs
Ready
Add to Source Control
Type here to search
ENG
10:47
22-01-2021


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Ex7_Session
Sign in
Debug
Any CPU
IIS Express
Live Share
Details1.cshtml
Details.cshtml
Employee.cs
EmployeeController.cs
Output
Inbox.cshtml
AccountController.cs
Login.cshtml
HomeController.cs
Ex7_Session
Ex7_Session.Controllers.EmployeeController
Details1()
public IActionResult Details()
{
ViewBag.Title = "Employee Details";
Employee employee = new Employee()
{
Id=101,
Name="raj",
City="Hyderabad",
Gender="Male"
};
return View(employee);
}
public IActionResult Details1()
{
ViewBag.Title = "Employee Details1";
List<Employee> employees = new List<Employee> {
new Employee{Id=101,Name="Raj",City="delhi",Gender="Male"},
new Employee{Id=102,Name="Rajee",City="delhi",Gender="Female"}
};
return View(employees);
}
}
}
120 %
No issues found
Ln: 25
Ch: 35
SPC
CRLF
Error List
Build succeeded
Add to Source Control
Type here to search
ENG
11:00
22-01-2021

```
@model IEnumerable<Ex7_Session.Models.Employee>

@{
    ViewData["Title"] = "Details1";
}

<h2>Details1</h2>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>

                @Html.DisplayNameFor(model => model.Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Name)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.City)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Gender)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
```

```
@foreach (var item in Model) {
    <tr>
      <td>
        @item.Id
        @Html.DisplayFor(modelItem => item.Id)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Name)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.City)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.Gender)
      </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 */ })
      </td>
    </tr>
}
  </tbody>
</table>
```

Advantages of using Strongly Typed View in ASP.NET Core MVC Application:

- We will get the following advantages when we use a strongly typed view in the ASP.NET Core MVC application.
  - It will provide compile-time error checking, as a result, we will get the intelligence support.
  - With intelligence support, the chances of mis-spelling the properties and making typographical errors are almost zero.
  - If we misspell the property name, then it comes to know at compile time rather than at runtime.
  - The best and preferred approach in ASP.NET Core MVC to pass data from a controller action method to a view is by using a strongly typed model object.
- In our example, we are still using ViewBag to pass the Header and Title from the Controller action method to the View. Then definitely the question that comes to yo mind is how we will pass the Header and Title to a strongly typed view. Well, in such scenarios we need to use a view specific model which is called View Model.

Thanks