

Episode 5

CONTROLLERS AND ROUTING IN ASP.NET CORE

swipe



Dhruv Mehta
@DhruvMehta1999



CONTROLLERS AND ATTRIBUTE- BASED ROUTING IN ASP.NET CORE

Learn how to create controllers and define routes efficiently!

1. What is a Controller?
2. How to create a Controller step by step
3. Understanding Attribute-Based Routing

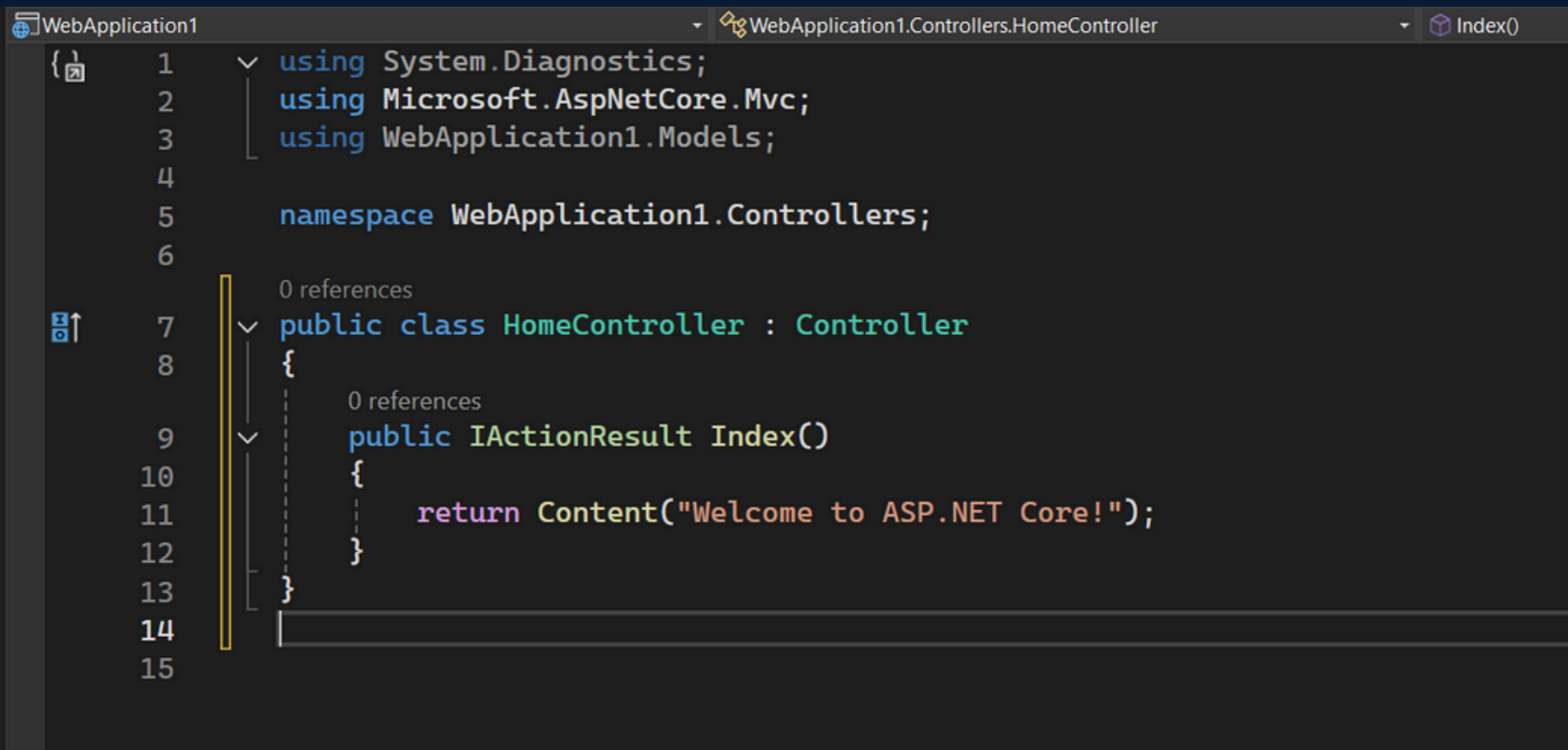
swipe



WHAT IS A CONTROLLER?

A Controller is a C# class that handles HTTP requests and returns responses.

1. It acts as a bridge between models (data) and views (UI).
2. Controllers contain action methods that process requests.



```
1  using System.Diagnostics;
2  using Microsoft.AspNetCore.Mvc;
3  using WebApplication1.Models;
4
5  namespace WebApplication1.Controllers;
6
7  public class HomeController : Controller
8  {
9      public IActionResult Index()
10     {
11         return Content("Welcome to ASP.NET Core!");
12     }
13 }
14
15
```

Requests to /home/index will execute Index()

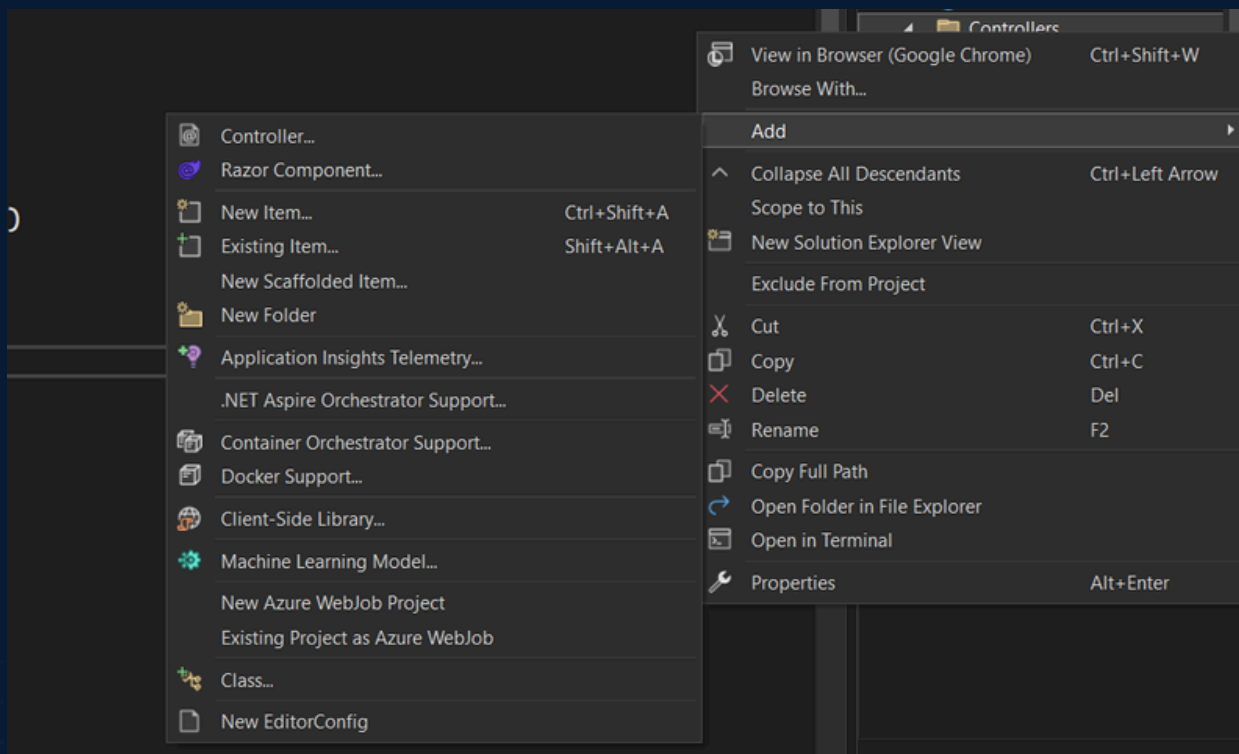
Dhruv Mehta
@DhruvMehta1999

swipe



STEPS TO CREATE A CONTROLLER

- Step 1: Open your ASP.NET Core project.
- Step 2: Navigate to the Controllers folder.
- Step 3: Right-click → Add → Controller → Empty.
- Step 4: Name it (e.g., ProductsController).
- Step 5: Write an action method inside the controller.



swipe



Dhruv Mehta

@DhruvMehta1999



HOW TO CONFIGURE CONTROLLER?

In order to use controllers, you need to add controllers as a service and you need to map them. We'll do this in our beloved Program.cs file.

```
{  
  1  var builder = WebApplication.CreateBuilder(args);  
  2  builder.Services.AddControllers();  
  3  var app = builder.Build();  
  4  
  5  app.MapControllers();  
  6  
  7  app.MapGet("/", () => "Hello World!");  
  8  
  9  app.Run();  
 10
```

builder.Services.AddControllers();

app.MapControllers();

We need to add these 2 lines, as we're telling our application that we'll be using controllers in our Web Application.

Dhruv Mehta

@DhruvMehta1999

swipe



WHAT IS ATTRIBUTE-BASED ROUTING?

1. Attribute-Based Routing allows defining routes directly inside controllers instead of in Program.cs.
2. It provides more flexibility and keeps route definitions closer to action methods.

```
public class HomeController : Controller
{
    [Route("person")]
    public IActionResult PersonInfo(Person person)
    {
        return Content($"Name: {person.PersonName}, Email: {person.Email}");
    }
}
```

Example of Attribute Routing

Now, accessing /person?PersonName=John&Email=john@example.com will return:
"Name: John, Email: john@example.com"

Dhruv Mehta

@DhruvMehta1999

swipe



WHAT'S NEXT

Since I have explained about how to define routing, I'm going to explain Attribute Routing in Depth in Next Episode.

Trust me it's very interesting.

Cya in next episode.

Dhruv Mehta

@DhruvMehta1999

swipe



WHAT'S NEXT

In Episode 5, we'll start with creating our first controller. And we'll learn about setting Routing in Controller

Cya in next Episode.

Dhruv Mehta

@DhruvMehta1999

swipe

