Episode 4

# INTRO TO MIDDLEWARE IN ASP.NET CORE

swipe

**Dhruv Mehta**
@DhruvMehta1999

# UNDERSTANDING MIDDLEWARE IN ASP.NET CORE

Welcome back to the ASP.NET Core Series! In this episode, we'll break down Middleware, the backbone of request processing in ASP.NET Core.

1. What is Middleware?
2. How Middleware Works?
3. Example with app.Use, app.Run, and app.MapGet

swipe

**Dhruv Mehta**
@DhruvMehta1999

# WHAT IS MIDDLEWARE IN ASP.NET CORE?

Imagine you walk into a shopping mall. Before entering a store, you go through different steps:

1. Security Check – Guards check your bags (Authentication).
2. Guide at Entrance – Helps you find the right store (Routing).
3. Billing Counter – Verifies your items before purchase (Authorization).
4. Exit Door – Allows you to leave once everything is checked (Response Processing).

Each step modifies or checks your request before allowing you to proceed.

**Dhruv Mehta**
@DhruvMehta1999

swipe

# WHAT IS MIDDLEWARE IN ASP.NET CORE?

Middleware is a set of building blocks that process requests and responses in an ASP.NET Core app.

Middleware runs in order and can:

1. Check security (Authentication)
2. Route requests to the correct place (Routing)
3. Handle errors (Exception Handling)
4. Log information (Logging)
5. Modify responses (Compression, CORS, etc.)

Think of middleware as a series of checkpoints that every request passes through before reaching its destination.

**Dhruv Mehta**

@DhruvMehta1999

swipe

# ADDING MIDDLEWARE IN PROGRAM.CS

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.Use(async (context, next) =>
{
    Console.WriteLine("Middleware 1: Request received");
    await next(); // Pass request to next middleware
    Console.WriteLine("Middleware 1: Response sent");
});

app.Use(async (context, next) =>
{
    Console.WriteLine("Middleware 2: Processing request");
    await next(); // Pass request to next middleware
    Console.WriteLine("Middleware 2: Processing response");
});

app.Run(async (context) =>
{
    await context.Response.WriteAsync("Hello from ASP.NET Core!");
});

app.Run();
```

**Dhruv Mehta**
@DhruvMehta1999

swipe

# EXPLAINING MIDDLEWARE

What happens?

1. Middleware 1 logs the request and passes it forward.
2. Middleware 2 processes it and passes it forward.
3. Final Middleware generates a response.

The response travels back through Middleware 2 → Middleware 1 → User.

**Dhruv Mehta**
@DhruvMehta1999

swipe

# IMPORTANCE OF MIDDLEWARE

Why is Middleware Important?

1. Controls how requests are handled.
2. Improves security by managing authentication & authorization.
3. Makes debugging easier with logging & error handling.
4. Optimizes performance by modifying requests & responses.

Middleware is the backbone of ASP.NET Core applications!

**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