



Deep Drive Into ASP.Net Core 2.1 (Day 5)

Presented by
DEBASIS SAHA



C# Corner MVP

| Team Lead | Author | Blogger | Speaker



debasis.ds@outlook.com



@debasiskolsaha



<https://github.com/debasis-saha>

About Me

Debasis Saha

- | Team Lead and Consultant
- | C# Corner MVP
- | DZone MVB

Email : debasis.ds@outlook.com

Twitter : @debasiskolsaha

LinkedIn : <https://www.linkedin.com/in/sahadebasis>

- ❖ **Authentications**
- ❖ **External Authentications**
- ❖ **Authorization**
- ❖ **Concept of Middleware**
- ❖ **Build – In Middleware**
- ❖ **Custom Middleware**
- ❖ **Middleware Pipe**
- ❖ **Q n A**

WHAT IS AUTHENTICATION ?

Authentication is the process of obtaining some sort of credentials from the users and using those credentials to verify the user's identity.

TYPES OF AUTHENTICATION

- ❖ **Windows Authentication**
- ❖ **Forms Authentication**
- ❖ **Passport Authentication**
- ❖ **Custom Authentication**

Authentication Templates in Core 2.1

- ❖ **Individual User Accounts**
- ❖ **Work or School Accounts**
- ❖ **Windows Authentication**

External Authentication

- ❖ Enables users to log in using OAuth 2.0 with credentials from external authentication providers.
- ❖ It Support any third party providers for authenticate.
- ❖ Some most common providers are –
 - ❖ Facebook
 - ❖ Twitter
 - ❖ Google
 - ❖ Microsoft



WHAT IS AUTHORIZATION ?

- ❖ **Authorization** is the process of allowing an authenticated user access to resources.
- ❖ Authorization is orthogonal and independent from authentication.
- ❖ Authentication is the process of ascertaining who a user is.
- ❖ Authentication may create one or more identities for the current user.

ROLE BASED AUTHORIZATION

- ❖ **Role Based Authorization** is an approach to restrict system access to authorise user.
- ❖ It depends on role-permission, user-role etc.

```
[Authorize(Roles = "Administrator")]  
public class AdministrationController : Controller  
{  
}
```

```
[Authorize(Roles = "HRManager, Finance")]  
public class SalaryController : Controller  
{  
}
```

CLAIM BASED AUTHORIZATION

- ❖ An identity is created it may be assigned one or more claims issued by a trusted party.
- ❖ An identity can contain multiple claims with multiple values and can contain multiple claims of the same type

```
services.AddAuthorization(options =>
{
    options.AddPolicy("EmployeeOnly", policy =>policy.RequireClaim("EmployeeNumber"));
});
```

```
[Authorize(Policy = "EmployeeOnly")]
public IActionResult VacationBalance()
{
    return View();
}
```

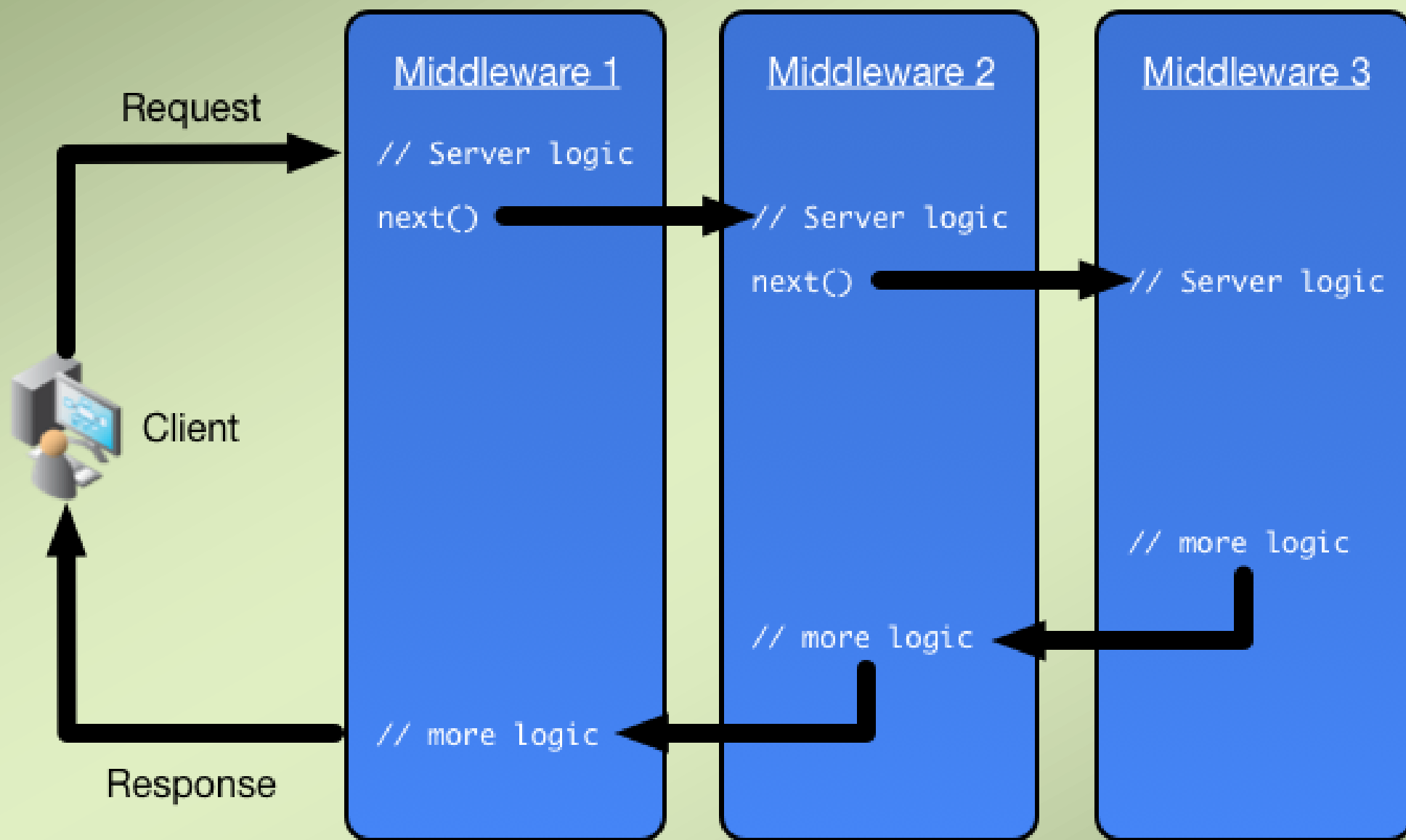
DEMO

On

Authentication & Authorizations

WHAT IS MIDDLEWARE ?

- ❖ **Middleware is software that's assembled into an app pipeline to handle requests and responses.**
- ❖ **Chooses whether to pass the request to the next component in the pipeline**
- ❖ **Request delegates are used to build the request pipeline. The request delegates handle each HTTP request.**
- ❖ **Request delegates are configured using Run, Map, and Use extension methods**



Middleware Order

- ❖ **The order for middleware components are added in the `Startup.Configure` method**
- ❖ **As per order sequence, middleware components are invoked on requests and the reverse order for the response.**
- ❖ **The order is critical for security, performance, and functionality.**

Build-In Middleware

- ❖ Exception Handling
- ❖ Static Files
- ❖ Authentication
- ❖ Session
- ❖ MVC

```
public void Configure(IApplicationBuilder app)
{
    app.UseExceptionHandler("/Home/Error");
    app.UseStaticFiles();
    app.UseIdentity();
    app.UseSession();
    app.UseMvcWithDefaultRoute();
}
```

DEMO

Middleware

THANK YOU For Attending the Session

Session Resource is Available at –

- **Presentations & Sample Code :-**

https://github.com/debasis-saha/Asp.Net_Core_Session