NDC { Oslo } 2023

Flexible Authorisation with ASP.NET Core

increment
TECHNOLOGY CREATIVES
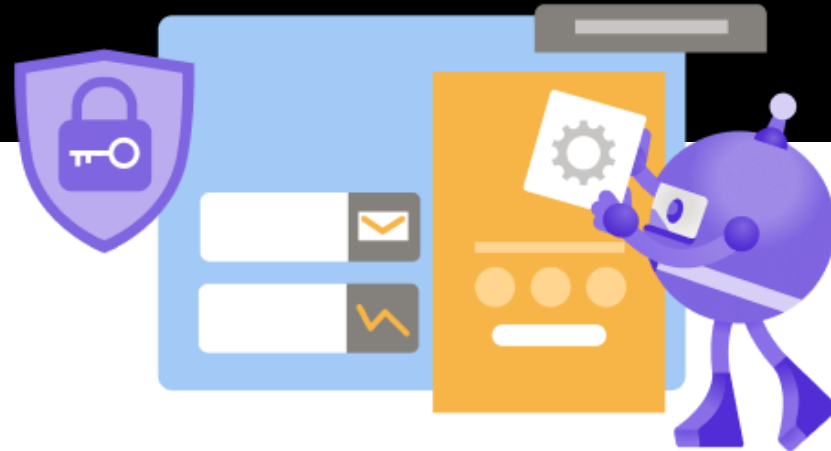
Jason Taylor

Solution Architect at Increment

Developer, Mentor, Speaker & Trainer

Microsoft MVP in Developer Technologies

Working with .NET since 1.0 in 2002

jasontaylordev

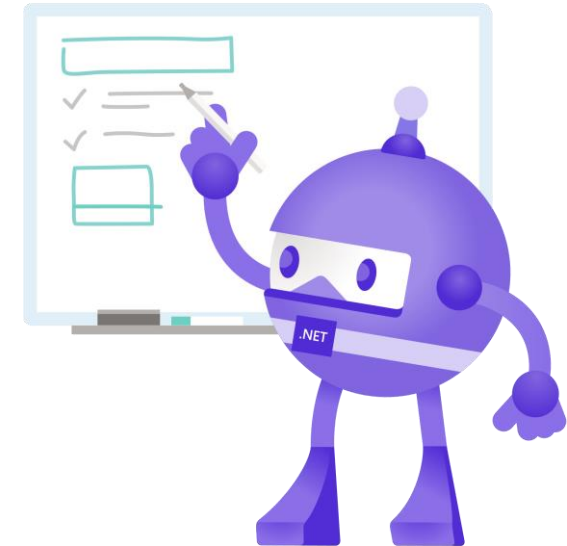github.com/jasontaylordev

jasontaylor.dev

youtube.com/jasontaylordev

Increment
TECHNOLOGY CREATIVES

Introduction

Basic Authorisation

Flexible Authorisation

Code Review

Summary

Background

- Authentication and authorisation

- Authorisation determines access to resources

- Includes defining and enforcing access control policies

- The focus for of this talk?

#NDCOslo @JasonTaylorDev

Authorisation in ASP.NET Core

Basic Authorisation

- Supports simple authorisation capabilities

- Includes sophisticated authorisation capabilities such as role-based, claims-based, and policy-based

- Easy to create custom authorisation policies

- Authorisation middleware also customisable

increment
TECHNOLOGY CREATIVES

Simple Authorisation

Basic Authorisation

- Apply [Authorize] to a controller, action, or Razor page

```csharp
[Authorize]
public class AccountController : Controller
{
    public ActionResult Login()
    {
    }


    public ActionResult Logout()
    {
    }
}
```

choose δifference

increment.inc

increment
TECHNOLOGY CREATIVES

Simple Authorisation

Basic Authorisation

- Apply [Authorize] to a controller, action, or Razor page

```csharp
public class AccountController : Controller
{
    public ActionResult Login()
    {
    }


    [Authorize]
    public ActionResult Logout()
    {
    }
}
```

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Simple Authorisation

Basic Authorisation

- Apply [Authorize] to a controller, action, or Razor page

```csharp
[Authorize]
public class AccountController : Controller
{

    [AllowAnonymous]
    public ActionResult Login()
    {
    }


    public ActionResult Logout()
    {
    }
}
```

choose δifference

increment.inc

increment
TECHNOLOGY CREATIVES

Basic Authorisation

- Role-based authorisation

```
[Authorize(Roles = "Administrators")]
public class AdministratorController : Controller
{
        public ActionResult Index()
        {
        }
}
```

increment
TECHNOLOGY CREATIVES

Authorisation with Roles

- Role-based authorisation

```csharp
[Authorize(Roles = "Administrators, Accounts")]
public class AccountsController : Controller
{
    public ActionResult Index()
    {
    }
}
```

increment
TECHNOLOGY CREATIVES

Authorisation with Roles

- Role-based authorisation

```csharp
[Authorize(Roles = "Administrators")]
[Authorize(Roles = "Accounts")]
public class AccountsController : Controller
{
        public ActionResult Index()
        {
        }
}
```

increment
TECHNOLOGY CREATIVES

Authorisation with Policies

- Policy-based authorisation

```
[Authorize(Policy = "EmployeesOnly")]
public class EmployeesController : Controller
{
    public ActionResult Index()
    {
    }
}`
```

increment
TECHNOLOGY CREATIVES

Basic Authorisation

- Policy-based authorisation

```
builder.Services.AddAuthorization(options =>
{
        options.AddPolicy("EmployeesOnly",
                    policy => policy.RequireClaim("EmployeeNumber"));
}
```

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Authorisation in Blazor WebAssembly

Basic Authorisation

- Supports basic authorisation capabilities

- Includes sophisticated authorisation capabilities such as role-based, claims-based, and policy-based

- Includes helpful authorisation components AuthorizeView and AuthorizeRouteView

- Authorisation is only used for UI/UX

increment
TECHNOLOGY CREATIVES

Simple Authorisation

- Apply [Authorize] to a routable page

```
@page "/"
@attribute [Authorize]

<PageTitle>Home</PageTitle>
```

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Simple Authorisation

Basic Authorisation

- Apply [AllowAnonymous] to a routable page

```
@page "/login"
@attribute [AllowAnonymous]

<PageTitle>Login</PageTitle>
```

increment
TECHNOLOGY CREATIVES

Authorisation with Roles

Basic Authorisation

- Role-based authorisation

```
@page "/counter"
@attribute [Authorize(Roles = "Administrators, Accounts")]

<PageTitle>Counter</PageTitle>
```

choose δifference

increment.inc

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Authorisation with Policies

Basic Authorisation

- Policy-based authorisation

```
@page "/users"
@attribute [Authorize(Policy = "EmployeesOnly")]

<PageTitle>Users</PageTitle>
```

increment
TECHNOLOGY CREATIVES

Authorisation with Views

Basic Authorisation

- Apply <AuthorizeView> to selectively display content

```
<AuthorizeView>
    <NavLink href="/">Home</NavLink>
</AuthorizeView>


<AuthorizeView Roles="Administrators, Accounts">
    <NavLink href="counter">Counter</NavLink>
</AuthorizeView>


<AuthorizeView Policy="EmployeesOnly">
    <NavLink href="users">Users</NavLink>
</AuthorizeView>
```

choose δifference

increment.inc

Page | 19

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Authorisation with Views

Basic Authorisation

- Apply <AuthorizeView> to selectively display content

```
<AuthorizeView>
    <NotAuthorized>
        <NavLink href="login">Login</NavLink>
    </NotAuthorized>
</AuthorizeView>
```

choose δifference

increment.inc

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Demo: Adding a new role

Basic Authorisation

- Adding a new role to an existing application

choose δifference

increment.inc

Flexible Authorisation with ASP.NET Core

Flexible Authorisation

- Easily add new roles and configure access control

- Easily reconfigure access control for existing roles

- Remove roles without impacting existing access control checks

- Easily view access control policies

- Support all of the above as standard application features

increment
TECHNOLOGY CREATIVES

Flexible Authorisation

- Adding a new role to an existing application

Basic vs. Flexible Approach

Flexible Authorisation

Basic

- Code changes

- Testing

- Build & deploy

- Bug fixes

- Documentation

Flexible

- Free time 🏝️

choose δifference

increment.inc

Page | 24

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES

Demo: Adding a new permission

Flexible Authorisation

- Adding a permission to support a new requirement

choose δifference

increment.inc

Overview

- Create necessary roles and permissions

- Assign permissions to roles as required

- Add permissions claim to authenticated user

- Add permissions-based access control as required

- Enforced by dynamic authorisation policies

increment
TECHNOLOGY CREATIVES

Managing Roles

- Roles define a logical grouping of users

- Administrators create new roles as required

- Administrators assign permissions to roles

- Role-based authorisation should be avoided

choose δifference

increment.inc

increment
TECHNOLOGY CREATIVES

Managing Permissions

Flexible Authorisation

- Permissions define granular access to resources

- Developers create permissions as necessary

- Permissions are not assigned directly to a user

- Permissions are assigned to a role

- Users inherit the permissions of any assigned roles

increment
TECHNOLOGY CREATIVES

Authorisation Engine

Flexible Authorisation

- Built entirely using policy-based authorisation

- Using a custom [Authorize] attribute, developers specify required permissions

- Required permissions are translated into a policy name

- Policy will be dynamically created using a custom policy provider

increment
TECHNOLOGY CREATIVES

Flexible Authorisation
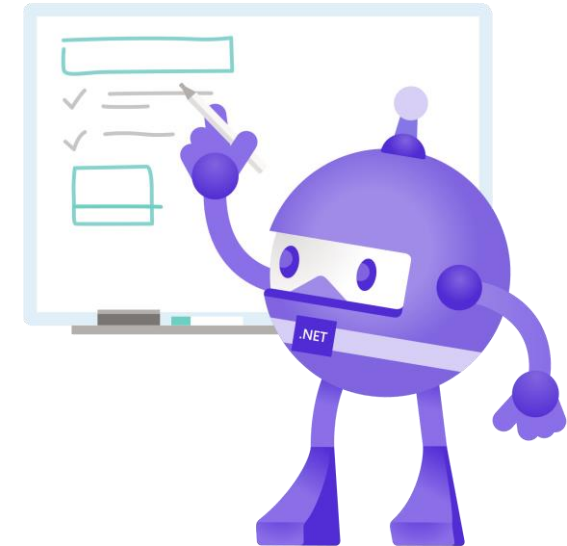
- An overview of the flexible authorisation engine

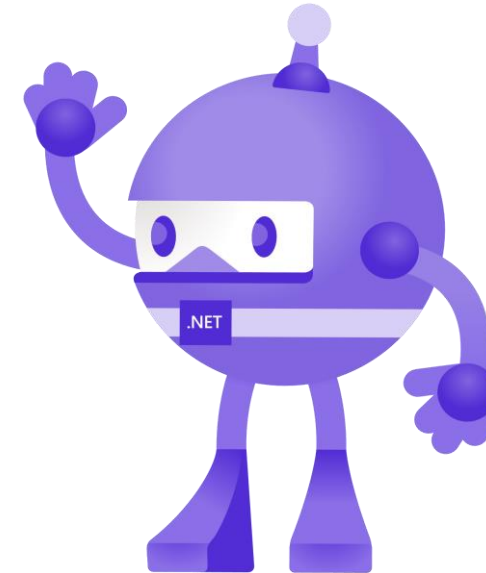Introduction

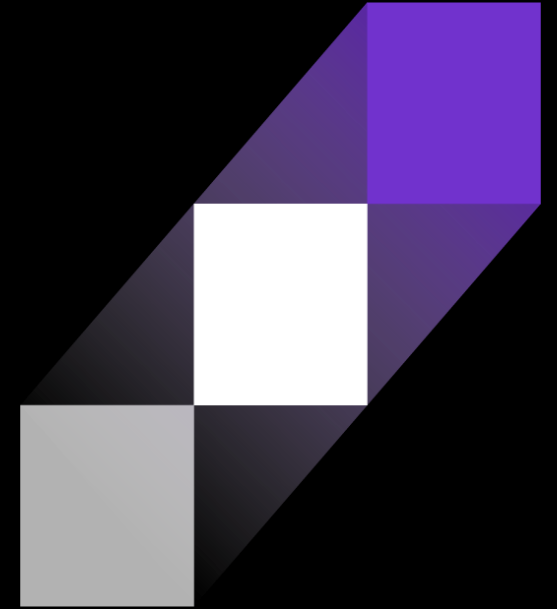Basic Authorisation

Flexible Authorisation

Code Review

Summary

choose δifference

increment.inc

Next Steps

Summary

- Access the code and slides on GitHub:

  https://github.com/jasontaylordev/flexible-aspnetcore-authorization

choose δifference

increment.inc

#NDCOslo @JasonTaylorDev

increment
TECHNOLOGY CREATIVES