Demystifying authentication in ASP.NET Core

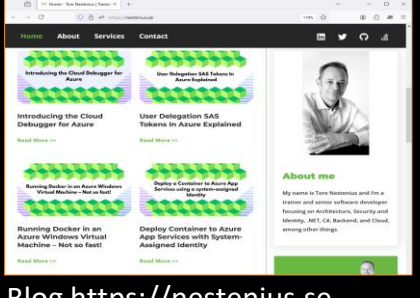
nestenius.se

https://nestenius.se

1

About Tore Nestenius

Work https://tn-data.se



Blog https://nestenius.se

https://meetup.com/net-skane

https://nestenius.se

2

My Courses

- **Fundamentals**
  - Containers and Kubernetes
  - Introduction to Git and GitHub
- **Architecture**
  - Modern Application Architecture
  - Service Communication - REST vs. GraphQL vs. gRPC
- **OpenID Connect and OAuth**
  - Introduction to OIDC and OAuth
  - Securing ASP.NET Using OIDC and IdentityServer
  - Identity Server in Production
- **Web**
  - Web Security Fundamentals
- **Cloud**
  - **Introduction to Azure for Developers**
- **.NET**
  - Applied Domain-Driven Design in .NET
  - ASP.NET Core Fundamentals
  - Asynchronous Programming in C#
  - Building ASP.NET Core APIs
  - C# Fundamentals
  - C# - Beyond the Fundamentals
  - C# Expert
  - TDD in .NET

https://nestenius.se

Important Notes!

- **Interrupt me!**
- **Ask questions!**

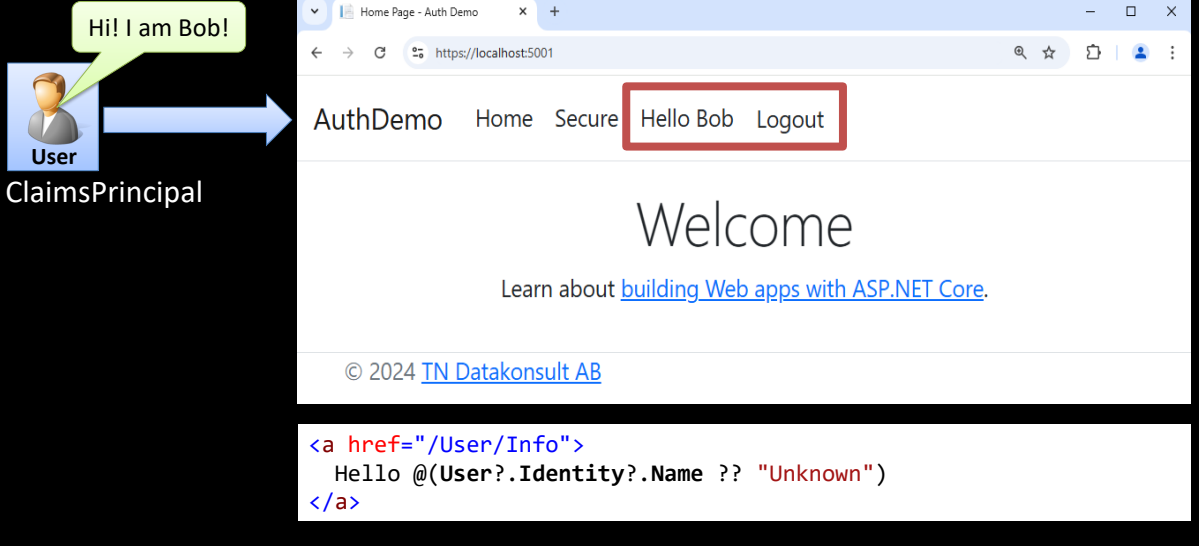
https://nestenius.se

What is the goal of this presentation?

https://nestenius.se

5

What does a page need to support this?

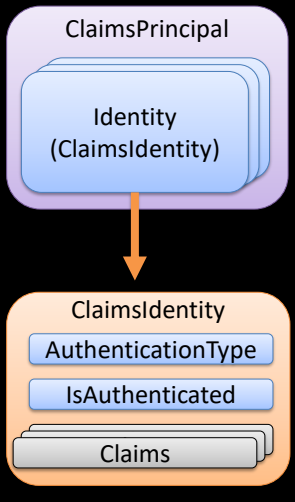



```
<a href="/User/Info">
  Hello @(User?.Identity?.Name ?? "Unknown")
</a>
```

https://nestenius.se

6

ClaimsPrincipal




```
public class ClaimsPrincipal : IPrincipal
{
    //List of identities
    List<ClaimsIdentity> _identities = new();

     ...
}
```

```
public class ClaimsIdentity : IIdentity
{
    //List of claims
    List<Claim> _instanceClaims = new();

    ...
}
```

https://nestenius.se

7

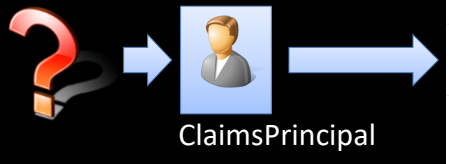
Claims

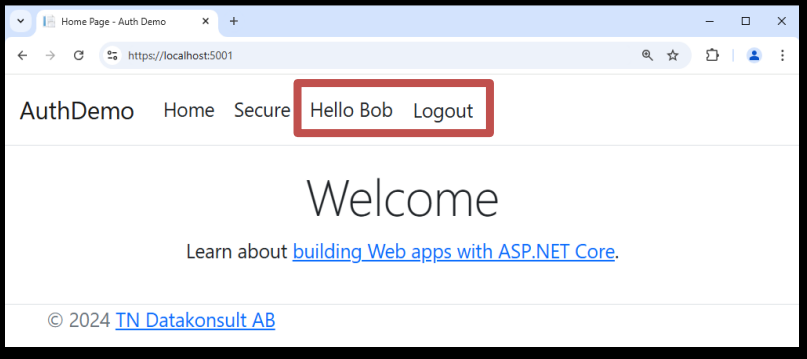
```
var claims = new List<Claim>()
       {
           new("sub","1234"),
           new("name","Bob"),
           new("email","bob@tn-data.se"),
           new("role","developer")
       };
```

https://nestenius.se

8

Who creates the ClaimsPrincipal?





https://nestenius.se

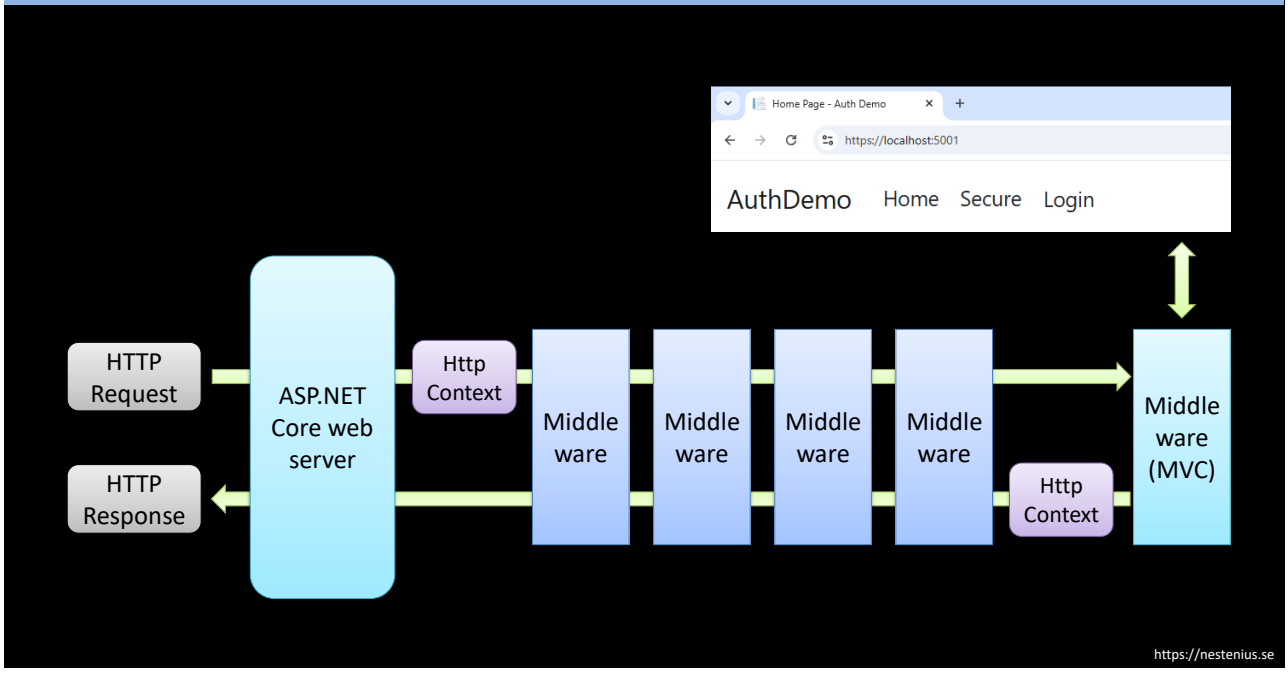
9

ASP.NET Core Request Pipeline

https://nestenius.se

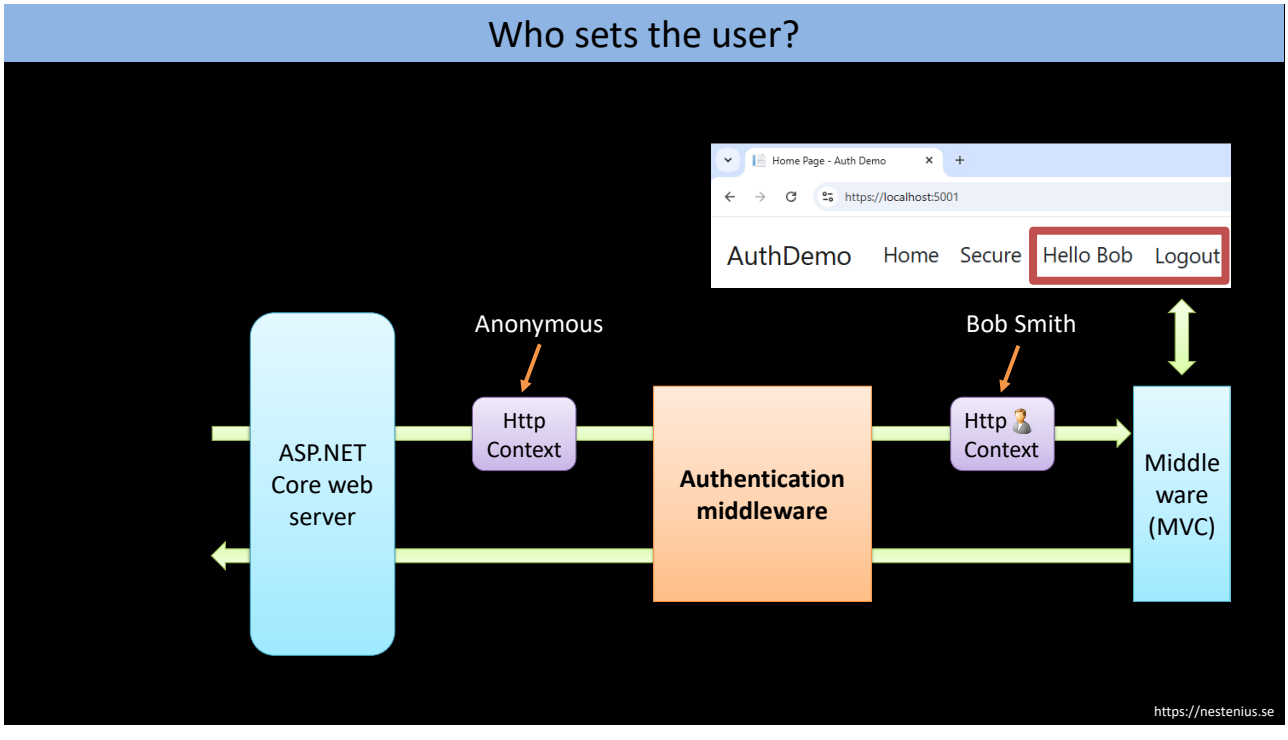
10

ASP.NET Core Request Pipeline



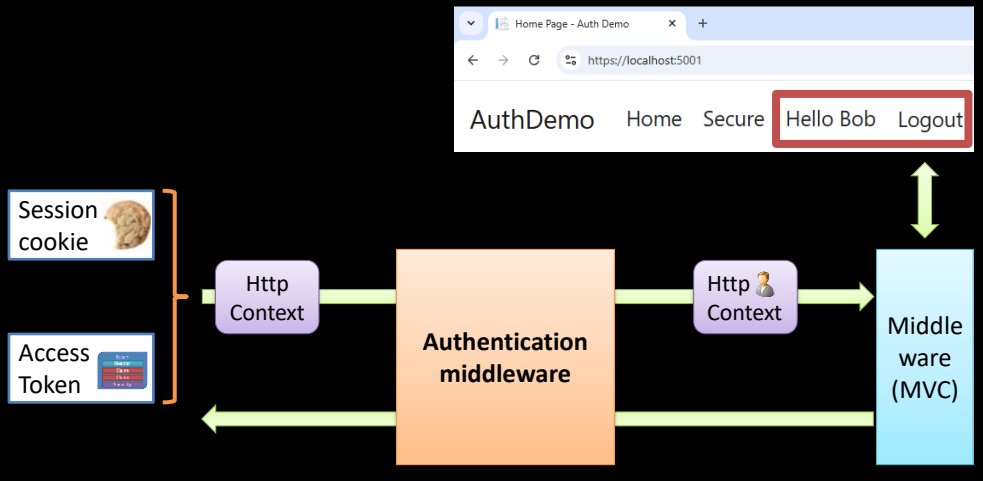
11

Who sets the user?



12

Where does the user data come from?



https://nestenius.se

13

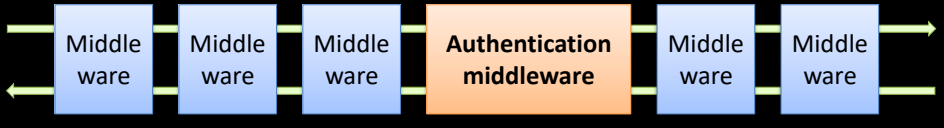
Live coding #1

```
builder.Services.AddAuthentication();

...

app.UseAuthentication();
```

https://nestenius.se

14

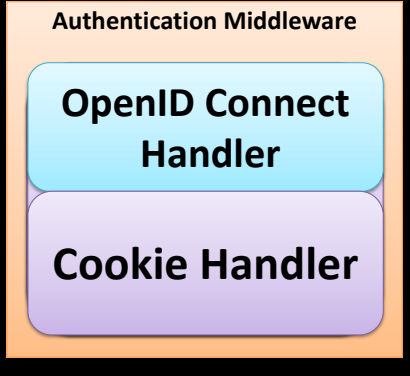
What is inside Authentication?



https://nestenius.se

15

Authentication Handlers




```
//Add the Cookie handler
builder.Services.AddAuthentication()
                .AddCookie();
```

```
// Add the JwtBearer handler (for APIs)
builder.Services.AddAuthentication()
                .AddJwtBearer();
```

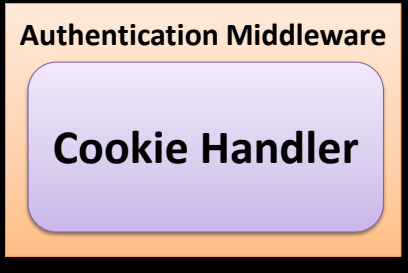
```
//Add the Cookie and OpenID-Connect handler
builder.Services.AddAuthentication()
                .AddCookie()
                .AddOpenIdConnect();
```

https://nestenius.se

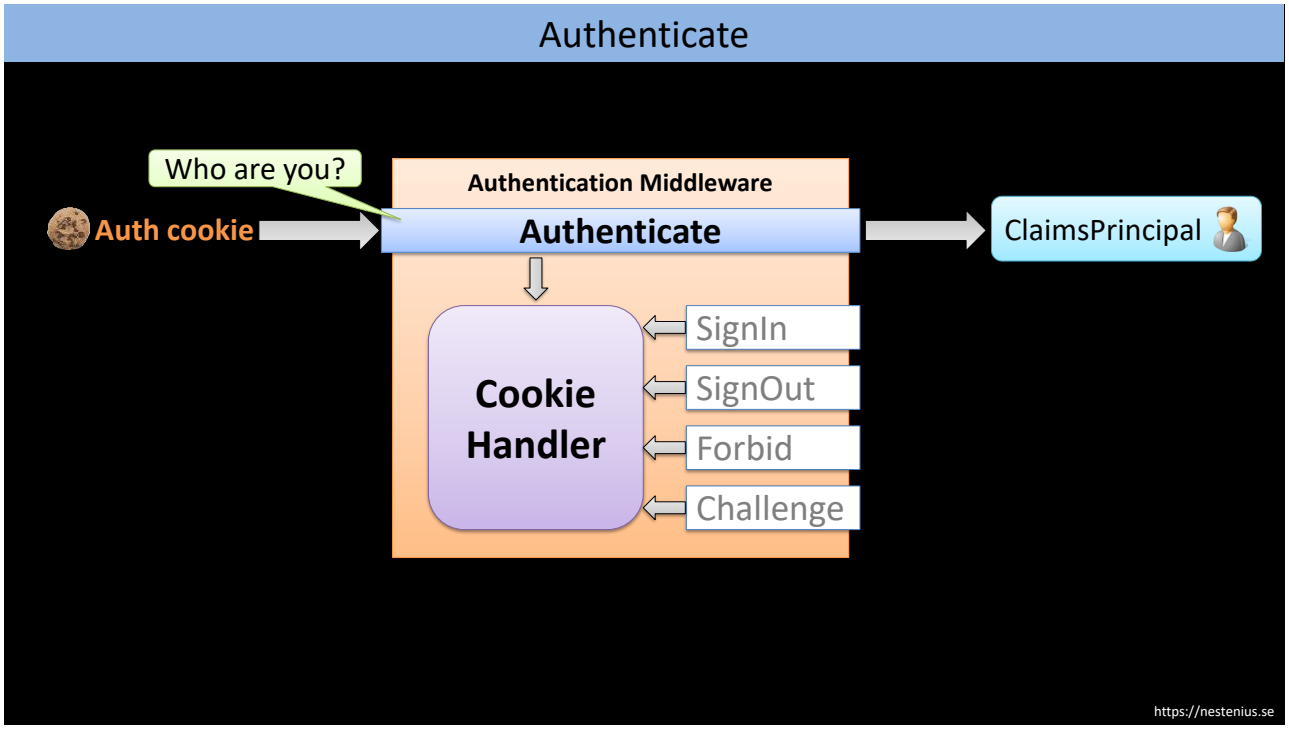
16

Live coding #2

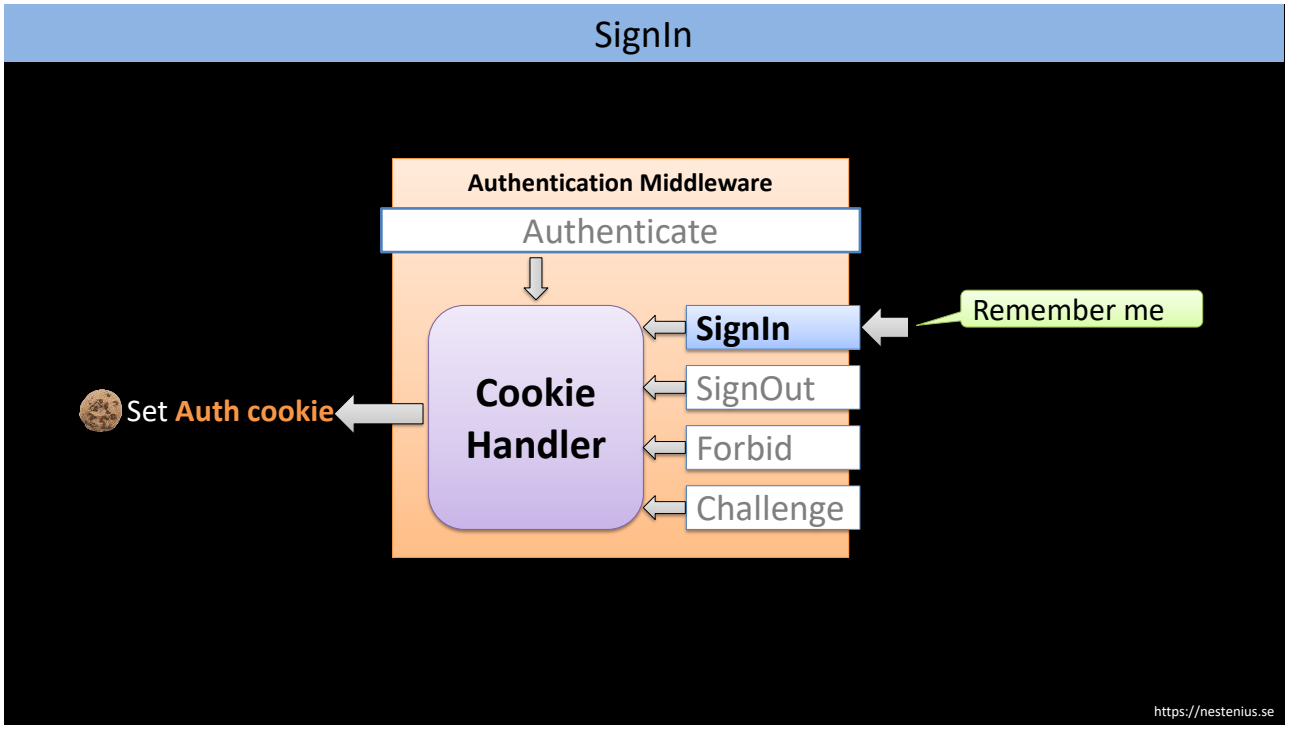
```
builder.Services.AddAuthentication()
                .AddCookie(o =>
                {
                    o.LoginPath = "/user/login";
                    o.LogoutPath = "/user/loggedOut";
                    o.AccessDeniedPath = "/user/AccessDenied";
                });
```



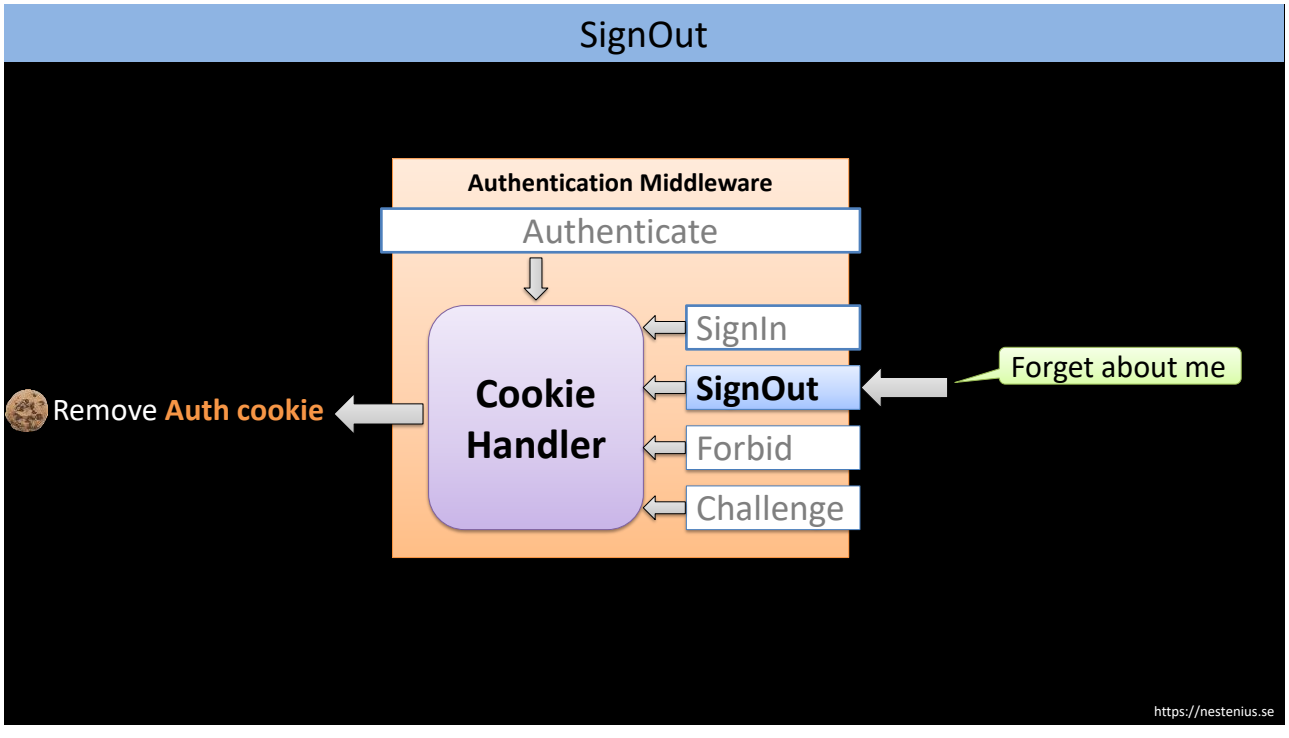
Authentication Operations


Authenticate
Who are you?
Auth cookie
Authentication Middleware
Authenticate
ClaimsPrincipal
Cookie Handler
SignIn
SignOut
Forbid
Challenge
https://nestenius.se

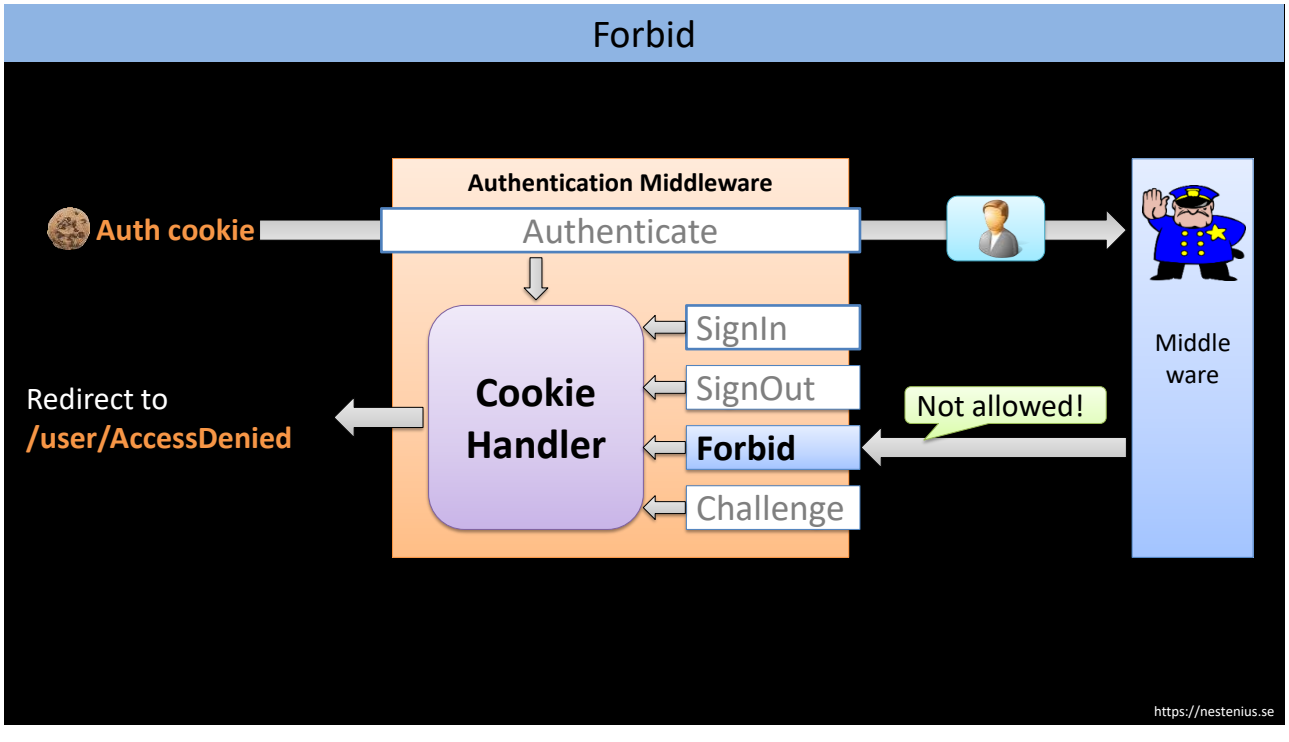
19


SignIn
Authentication Middleware
Authenticate
Remember me
Set Auth cookie
Cookie Handler
SignIn
SignOut
Forbid
Challenge
https://nestenius.se

20

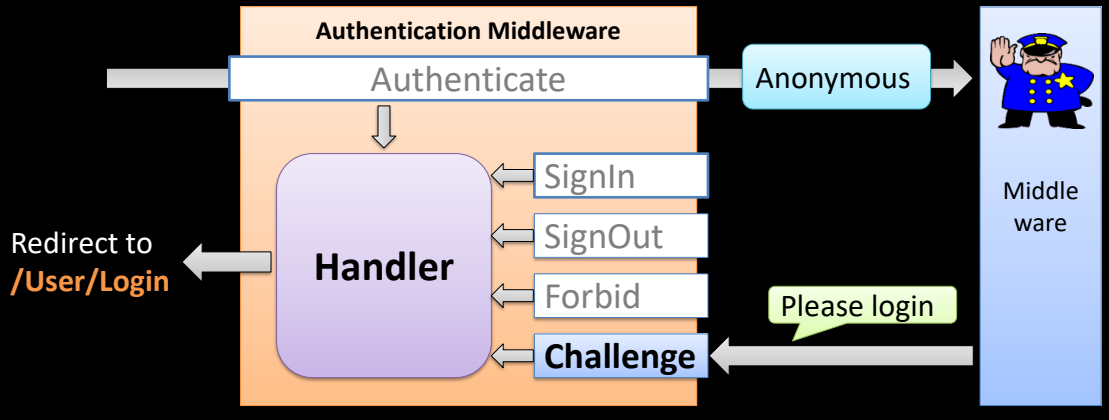

SignOut
Authentication Middleware
Authenticate
Cookie Handler
SignIn
SignOut
Forbid
Challenge
Forget about me
Remove Auth cookie
https://nestenius.se

21


Forbid
Authentication Middleware
Authenticate
Auth cookie
Cookie Handler
SignIn
SignOut
Forbid
Challenge
Redirect to
/user/AccessDenied
Not allowed!
Middle ware
https://nestenius.se

22

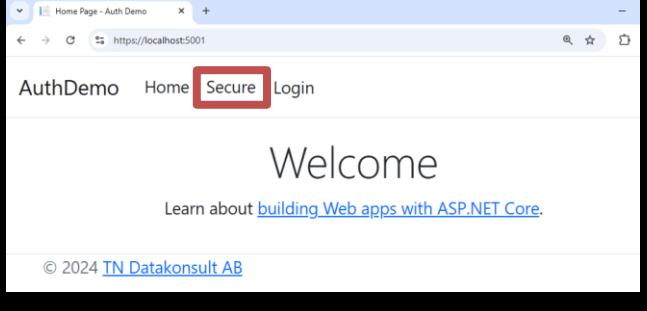
Challenge



https://nestenius.se

23

Live coding #3



https://localhost:5001/Secure

```
public async Task<IActionResult> Index()
{
    if (User.Identity.IsAuthenticated == false)
    {
        await HttpContext.ChallengeAsync();
    }

    return View();
}
```

https://nestenius.se

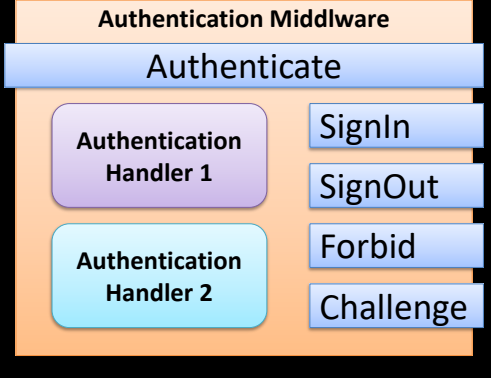
24

Schemes!

https://nestenius.se

25

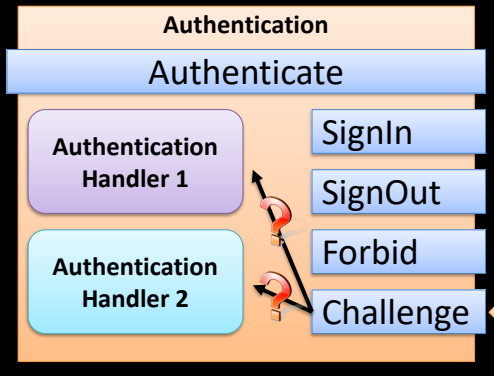
Schemes



https://nestenius.se

26

Schemes

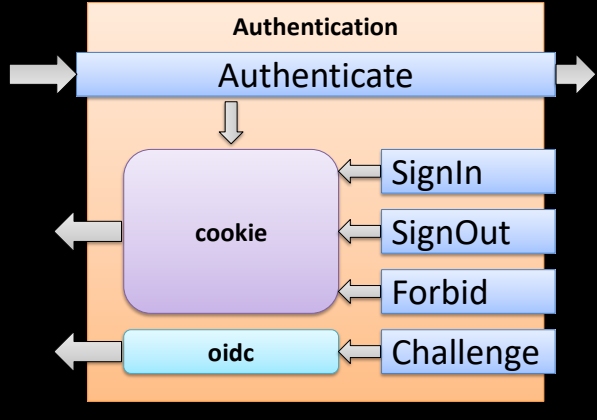



```
public async Task<IActionResult> Index()
{
    if (User.Identity.IsAuthenticated == false)
    {
        await HttpContext.ChallengeAsync();
    }

    return View();
}
```

https://nestenius.se

Schemes

```
builder.Services.AddAuthentication(o =>
{
    o.DefaultAuthenticateScheme = "cookie";
    o.DefaultChallengeScheme = "oidc";
    o.DefaultForbidScheme = "cookie";
    o.DefaultSignInScheme = "cookie";
    o.DefaultSignOutScheme = "cookie";
})
.AddCookie("cookie", opt =>
{
 ...
})
.AddOpenIdConnect("oidc", opt =>
{
  ...
});
```



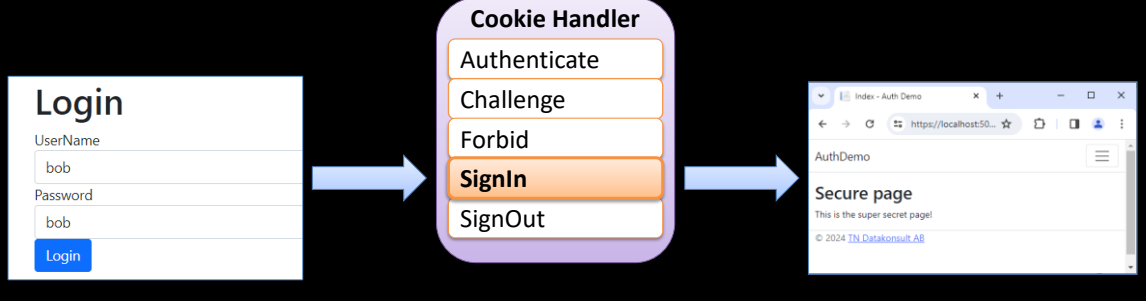
https://nestenius.se

Live coding #4

```
builder.Services.AddAuthentication(o =>
{
    o.DefaultScheme = "cookie";
})
.AddCookie("cookie", o =>
{
    o.LoginPath = "/user/login";
    o.LogoutPath = "/user/loggedOut";
    o.AccessDeniedPath = "/user/AccessDenied";
});
```

https://nestenius.se

SignIn the user



https://nestenius.se

Validate the username and password

```
[HttpGet]
public IActionResult Login(string ReturnUrl)
{
    return View(new LoginModel() { ReturnUrl = ReturnUrl });
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task Login(LoginModel loginCredentials)
{
    //...
}
```

```
public class LoginModel
{
    public string UserName  { get; set; };
    public string Password  { get; set; };
    public string ReturnUrl { get; set; };
}
```


Create the list of claims

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task Login(LoginModel loginCredentials)
{

    //1. Validate username + password

    //2. Load the claims for this user from the DB
    var claims = new List<Claim>()
                {
                    new("sub","1234"),
                    new("name","Bob"),
                    new("email","bob@tn-data.se"),
                    new("role","developer")
                };

    //...

}
```

Create an Identity + ClaimsPrincipal

```
//1. Validate username + password

//2. Load the claims for this user from the DB
var claims = new List<Claim>()
    {
        new("sub","1234"),
        new("name","Bob"),
        new("email","bob@tn-data.se"),
        new("role","developer")
    };

//3. Create an Identity for these claims
var identity = new ClaimsIdentity(claims: claims,
                                  authenticationType: "pwd",
                                  nameType: "name",
                                  roleType: "role");

//4. Create the principal based on the users Identity
var principal = new ClaimsPrincipal(identity);
```

Auth types
pwd
External
Windows
Jwt
Cookie
api_key
NTLM
Digest
Basic
Negotiate
…

https://nestenius.se

33

SignIn Principal

```
var principal = new ClaimsPrincipal(identity);

var prop = new AuthenticationProperties()
{
    RedirectUri = loginCredentials.ReturnUrl,
    Items =
    {
        { "IpAddress", "192.168.0.3" },
        { "ComputerName", "MyComputer" },
        { "ApiKey", "Summer2024!!" }
    }
};

await HttpContext.SignInAsync(scheme: "cookie",
                              principal: principal,
                              properties: prop);
```

https://nestenius.se

34

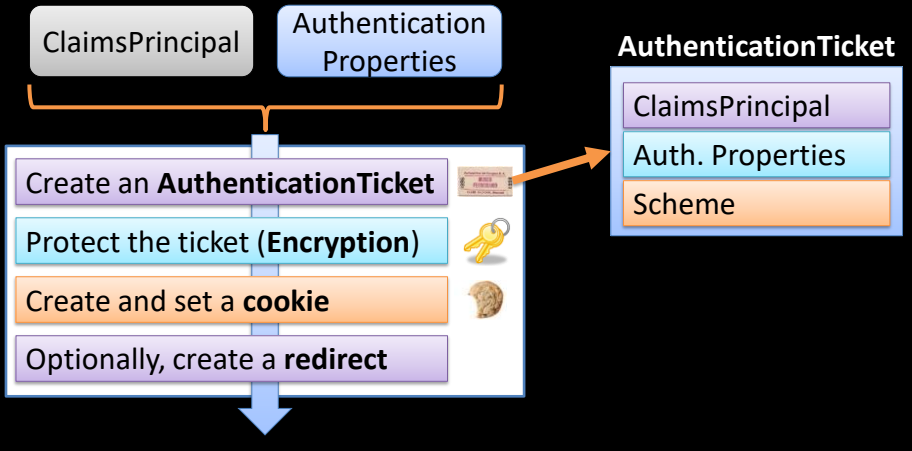
Live coding #5

```
//1. Validate username + password

//2. Load the claims for this user from the DB
var claims = new List<Claim>()
{
    new("sub","1234"),
    new("name","Bob"),
    new("email","bob@tn-data.se"),
    new("role","developer")
};

var identity = new ClaimsIdentity(claims: claims,
                                  authenticationType: "pwd",
                                  nameType: "name",
                                  roleType: "role");

var principal = new ClaimsPrincipal(identity);

var prop = new AuthenticationProperties()
{
    RedirectUri = "/",
    Items =
    {
        { "IpAddress", "192.168.0.3" },
        { "ComputerName", "MyComputer" },
        { "ApiKey", "Summer2024!!" }
    }
};

await HttpContext.SignInAsync(scheme: "cookie",
                              principal: principal,
                              properties: prop);
return LocalRedirect("/");
```


What happens when we call SignInAsync?

```
await HttpContext.SignInAsync(scheme: "cookie",
                              principal: principal,
                              properties: prop);
```

SignInAsync inside the Cookie Handler



https://nestenius.se

37

What is inside the authentication cookie?

```
Set-Cookie: .AspNetCore.cookie=CfDJ8IgPXRNAZH1EkNA0dd3_JvtpVOohM43sH8lB8MW4
2T1L57tPORRWmJu8svjUYUwIrYUOW4xoOikClOR3H87teUK4MYy58NBBAsjc8RDRWhKO6JVzOHuHW1eNSfu
nLJ_OOObZ1y6kYlF52lkzI8cw8VLPzG4zm33hoynL2JHLTCoWbugN-3kyOLrUSyVJdotB1ANGcvBT-
jz2rAFuOeUbzCdXXDjm98YzW3E99QffLamD1LlrKe7MX1y31NWdxzQ39m4WmGwUNa3bOiHoyDaeSKJvifmz
lMSWT_8o9x4AUtzC6_whIOfPVHXhYkwCXGTwtpIYeb_KOGuAvidb3S3tTkK4m3LHcf5Fx9ajfbE8RC_5FLO
sPxbQiQcF3KGmIUPOdnHmtK7MHczg4UR-OgBh_TA_sKyMoPy9Ak9sa4P-XvMWWyssEkOOzxfHbi6F
VWbq5CNDe6W1QZG6z5PwtwGsVmx4vK8C3_4b9r-HU; path=/; secure; samesite=lax; httponly
```

https://nestenius.se

38

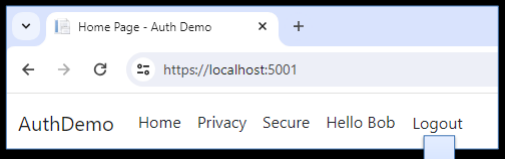
Live coding #6

```
.AddCookie(o =>
{
    o.DataProtectionProvider = new MyDataProtector();
});
```

```
public class MyDataProtector : IDataProtector
{
    public IDataProtector CreateProtector(string purpose)
    {
        return new MyDataProtector();
    }
    public byte[] Protect(byte[] plaintext)
    {
        return plaintext;
    }
    public byte[] Unprotect(byte[] protectedData)
    {
        return protectedData;
    }
}
```


SignOut

SignOut




```
public async Task Logout()
{
    //Sign out from this specific scheme
    await HttpContext.SignOutAsync("cookie");
}
```


```
Set-Cookie: .AspNetCore.cookie=; expires=Thu, 01 Jan 1970 00:00:00 GMT; ...
```

https://nestenius.se

41

Live coding #7

```
[HttpPost]
public async Task Logout()
{
    await HttpContext.SignOutAsync("cookie");
}
```

https://nestenius.se

42

LogoutPath

Why are we not redirected?

```
await HttpContext.SignOutAsync("cookie");
```

```
builder.Services.AddAuthentication()
                .AddCookie(o =>
                {
                    ...
                    o.LogoutPath = "/user/LoggedOut";
});
```

```
HTTP/1.1 200 OK
Content-Length: 0
Set-Cookie: .AspNetCore.cookie=; expires=Thu, 01 Jan 1970 00:00:00 GMT;...
```


Live coding #8

```
<form action="/User/Logout" method="post">
  <button type="submit">Logout</button>
</form>
```

```
<form action="/User/Logout?ReturnUrl=/User/LoggedOut" method="post">
  <button type="submit">Logout</button>
</form>
```

Using ReturnUrl

/User/Logout?ReturnUrl=/User/LoggedOut

```
builder.Services.AddAuthentication()
                .AddCookie(o =>
                {
                    ...
                    o.LogoutPath = "/user/LoggedOut";
                    ...
                });
```

```
HTTP/1.1 200 OK
Content-Length: 0
Set-Cookie: .AspNetCore.cookie=; expires=Thu, 01 Jan 1970 00:00:00 GMT;...
```

https://nestenius.se

Live coding #9

/User/Logout?ReturnUrl=/User/LoggedOut

```
.AddCookie(o =>
{
   ...
  o.LogoutPath = "/user/LoggedOut";
   ...
});
```

```
.AddCookie(o =>
{
   ...
   o.LogoutPath = "/user/LogOut";
   ...
});
```

```
HTTP/1.1 302 Found
Location: /User/LoggedOut
Set-Cookie: .AspNetCore.cookie=; expires=Thu, 01 Jan 1970 00:00:00 GMT;...
```

https://nestenius.se

Alternatives

```
[HttpPost]
public async Task Logout()
{
    var properties = new AuthenticationProperties()
    {
        RedirectUri = "/User/LoggedOut"
    };

    await HttpContext.SignOutAsync("cookie", properties);
}
```

```
[HttpPost]
public async Task<IActionResult> Logout()
{
    await HttpContext.SignOutAsync("cookie");

    return LocalRedirect("/User/LoggedOut");
}
```

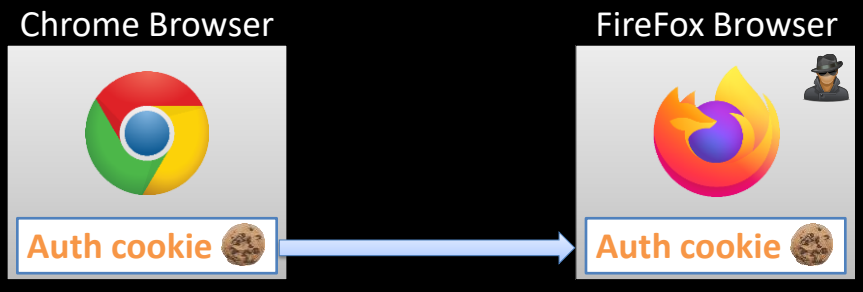
https://nestenius.se

Are we really signed out?

```
[HttpPost]
public async Task Logout()
{
    await HttpContext.SignOutAsync("cookie");
}
```

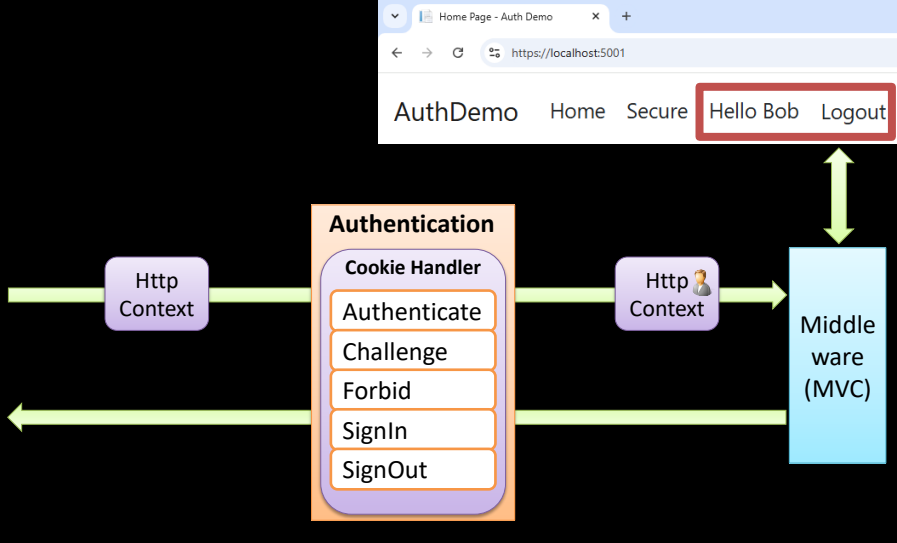
https://nestenius.se

Live coding #10



https://nestenius.se

Summary



https://nestenius.se

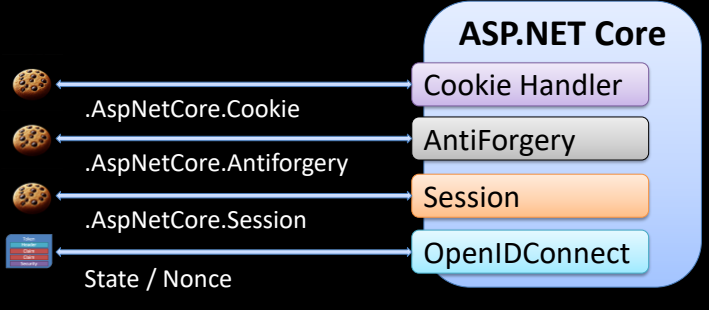
Data Protection

https://nestenius.se

51

Data Protection

ASP.NET Core issues several **sensitive items**, including:



It uses the **Data Protection API** to secure this

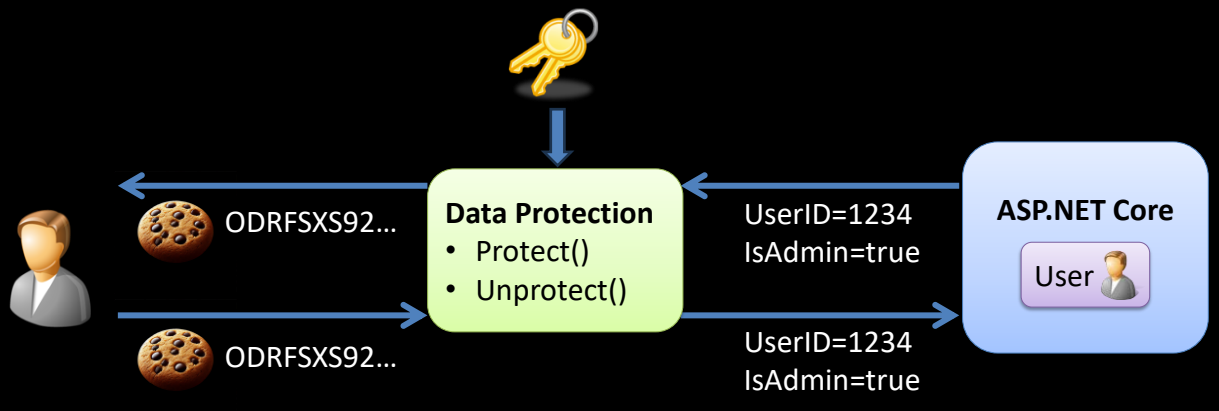
How does this work?

https://nestenius.se

52

Data Protection

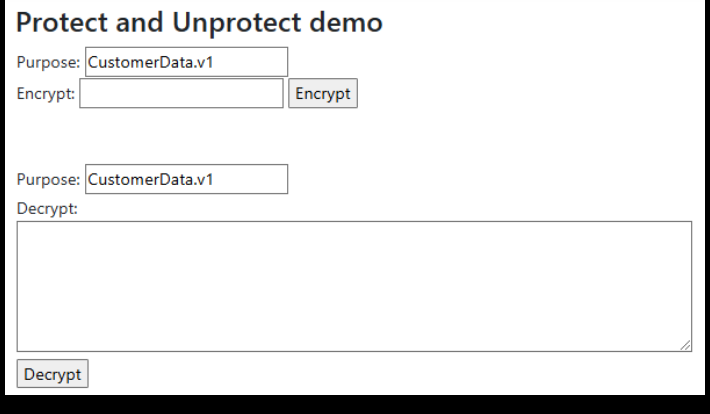
The data is secured using **encryption**



https://nestenius.se

53

Demonstration



https://nestenius.se

54

Data Protection in Production

```
Key {} was not found in the key ring. Unprotect operation cannot proceed.

cookie was not authenticated. Failure message: Unprotect ticket failed
```

https://nestenius.se

Data Protection

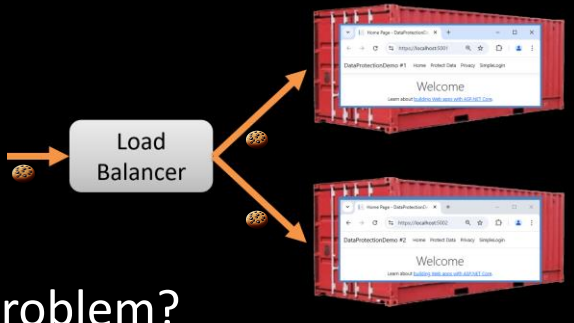
Example architecture

https://nestenius.se

Data Protection

What can go wrong here?

- Users lose their sessions after redeployment
- Cookies aren’t recognized across instances
- Key management
- …

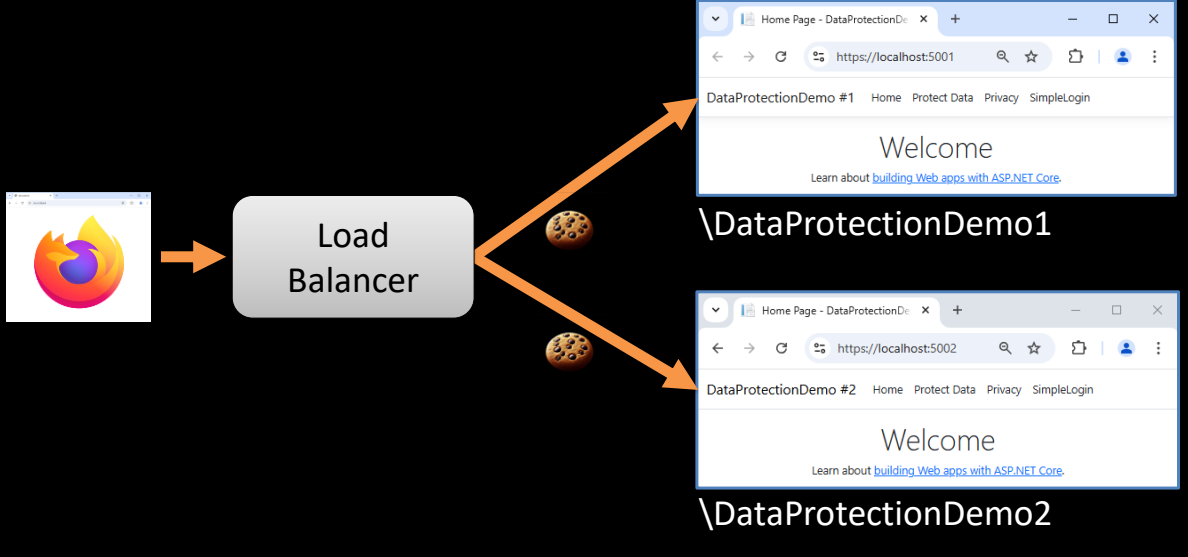


What is the problem?

https://nestenius.se

57

Demonstration – Cookies across service instances

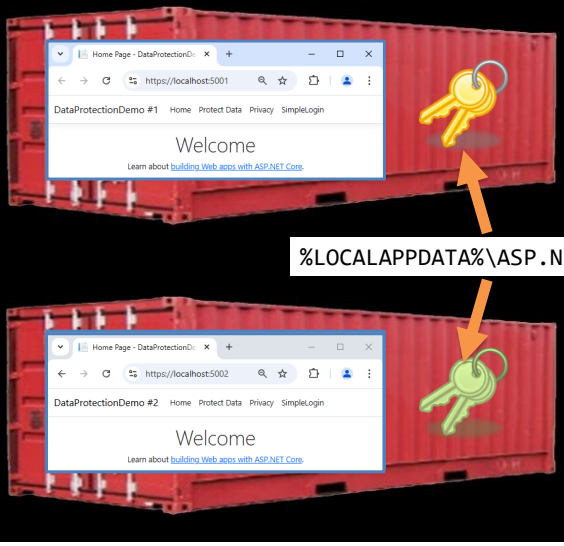


https://nestenius.se

58

Data Protection

By default, a **key** and a **key-ring** is created at startup

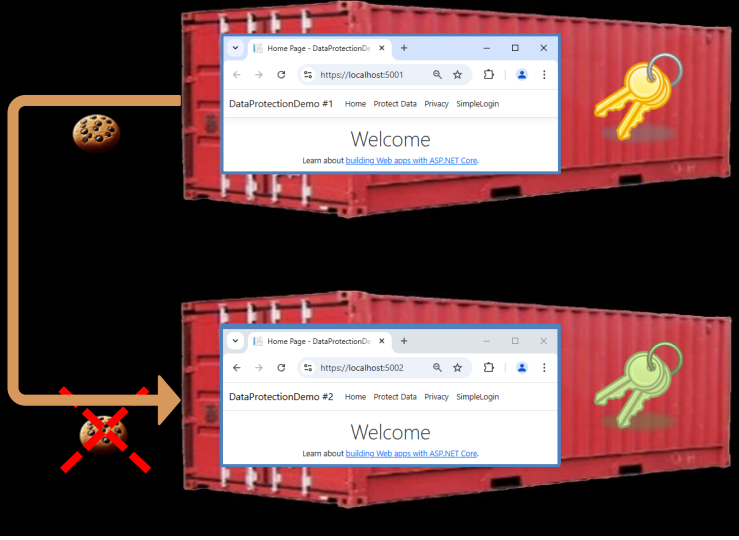


https://nestenius.se

59

Data Protection

Cookies will be rejected if the expected key is not found

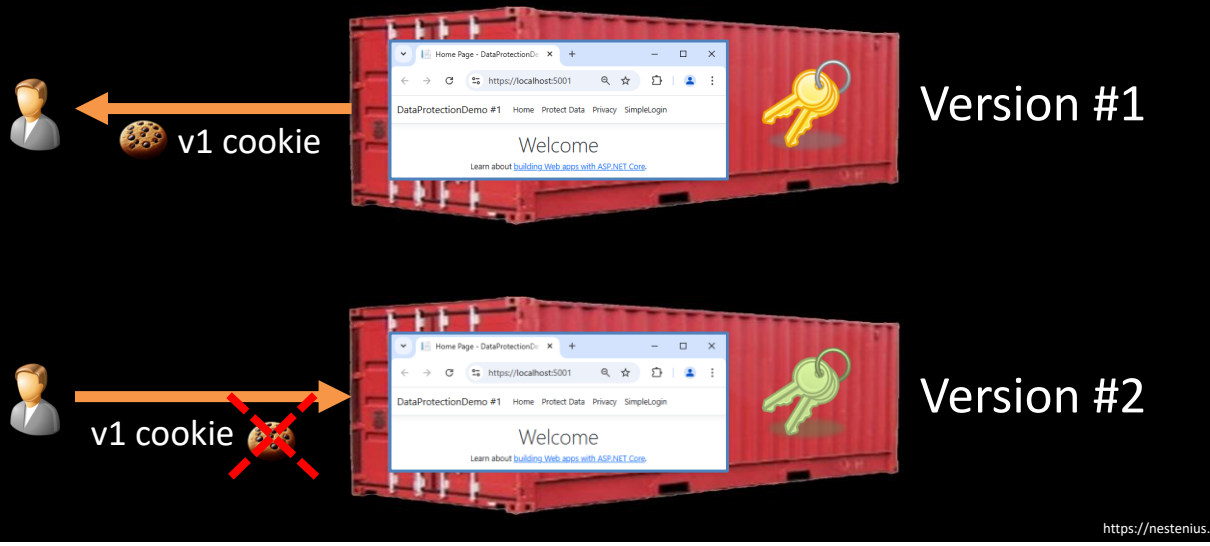


https://nestenius.se

60

Data Protection

Cookies will be rejected if the key changes after redeploy

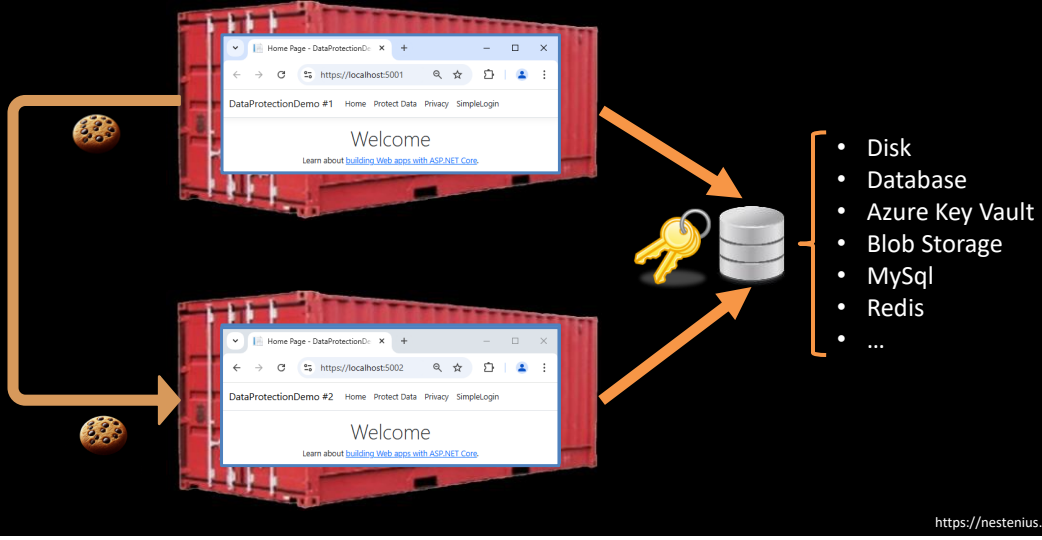


https://nestenius.se

61

Data Protection

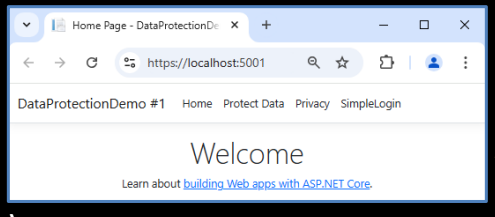
The **key-ring** needs to be persisted in a shared storage



https://nestenius.se

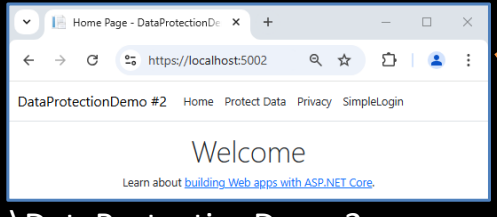
62

Demonstration - PersistKeysToFileSystem



\DataProtectionDemo1

```
ŵắs lêỳRắţḥ   ŋêx DîsêçţǰsỳÍŋğǰ   C  Cǫŋğ ḳêỳş

čụîľđês Şêsŵîçęş
        AđđDắţǻRsǫţǰêçţǰîǫŋ
        RêssîşţǰḳêỳşŢǫGîľêŞỳşţǰêŋ lêỳRắţḥ
```



\DataProtectionDemo2

```
ŵắs lêỳRắţḥ   ŋêx DîsêçţǰsỳÍŋğǰ   C  Cǫŋğ ḳêỳş

čụîľđês Şêsŵîçęş
        AđđDắţǻRsǫţǰêçţǰîǫŋ
        RêssîşţǰḳêỳşŢǫGîľêŞỳşţǰêŋ lêỳRắţḥ
```

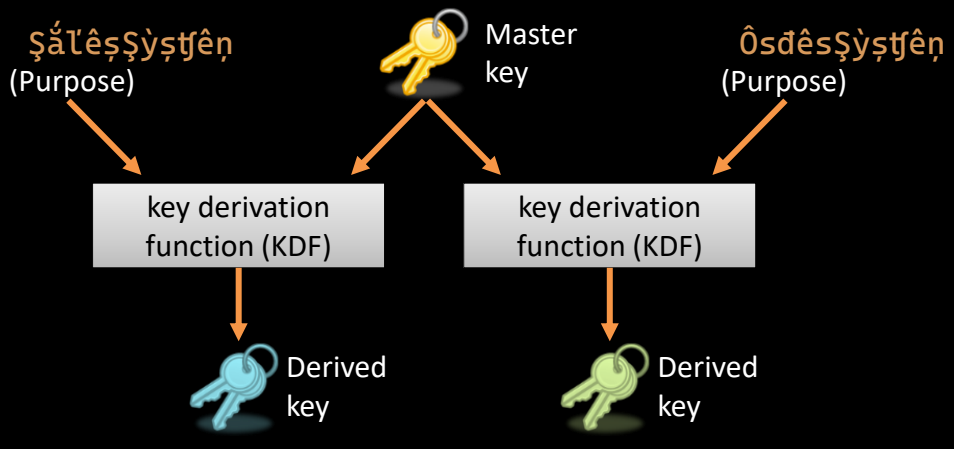

63

Data Protection Purpose

64

Data Protection Purpose

The **purpose** is used to create isolation



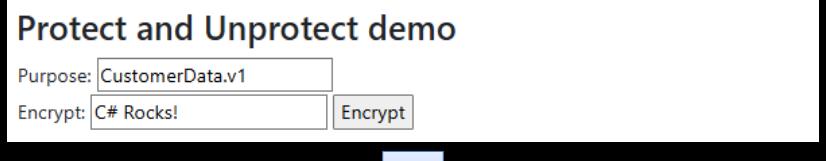
https://nestenius.se

65

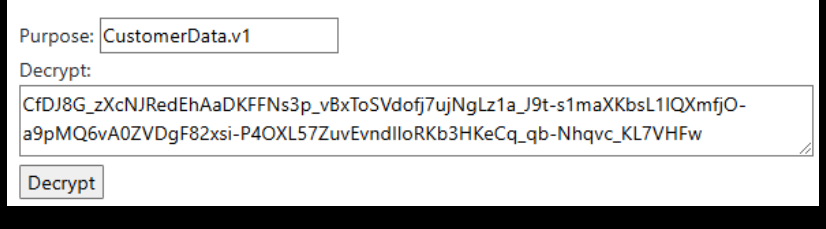
Demonstration - Purpose

Why does this not work?

Application #1



Application #2



https://nestenius.se

66

Data Protection Purpose

The App path is part of all purposes in ASP.NET Core:

```
Performing protect operation to key {...} with purposes (
  - 'C:\Conf\Demo - Data Protection\DataProtectionDemo1\',
  - 'SessionMiddleware').

Performing protect operation to key {...} with purposes (
  - 'C:\Conf\Demo - Data Protection\DataProtectionDemo1\',
  - 'Microsoft.AspNetCore.Authentication.Cookies.CookieAuthenticationMiddleware',
  - 'cookie',
  - 'v2').

Performing protect operation to key {...} with purposes (
  - 'C:\Conf\Demo - Data Protection\DataProtectionDemo1\',
  - 'CustomerData.v1').
```

https://nestenius.se

67

Demonstration - SetApplicationName

```
ŵắs lêỳRắţḫ   ŋêx DîsêçţǰŝỳÍŋğǰ   C  Cǫŋğ Ķêỳş

čụîľđês Şêsŵîçêş AđđDắţǰắRsǰţǰêçţǰîǫŋ
                Rêssịşţǰķêỳş[ǰGîľêŞỳşţǰêŋ lêỳRắţḫ
                ŞêţǰAřřľîçắţǰîǫŋŅắŋê  ŇỳAřřľîçắţǰîǫŋ
```

Sets the purpose

https://nestenius.se

68

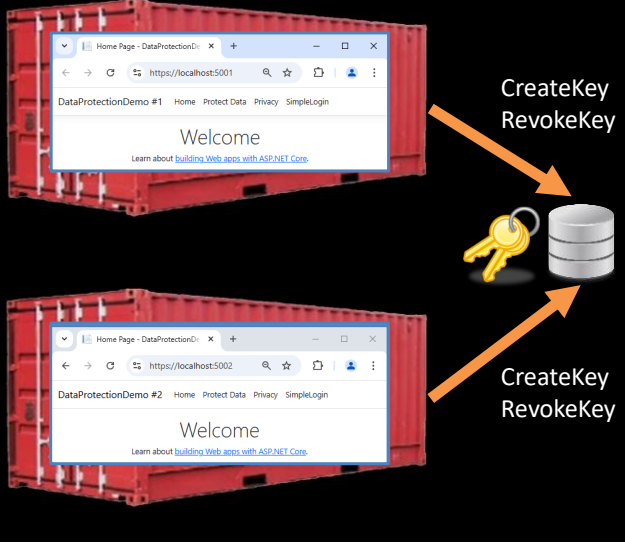
Key Management Problems

https://nestenius.se

69

Key Management Problems

In this setup, both services wants to manage the keys



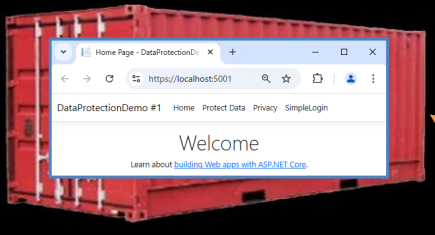
https://nestenius.se

70

Key Management Problems

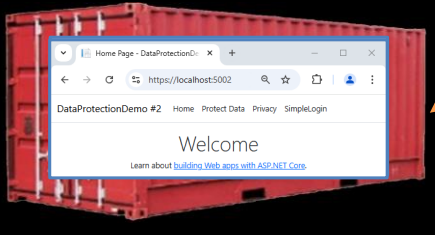
You can prevent it from generating new keys

```
Rsîņăsỳ
čụîľđês Şêsŵîçêş AđđDăţǧăRsộţŷêçţŷîộŋ
    ŞêţJAřřľîçăţŷîộŋŊăņê  ŇỳAřřľîçăţŷîộŋ
```



ReadKey
CreateKey
RevokeKey

```
Şêçộŋđăsỳ
čụîľđês Şêsŵîçêş AđđDăţǧăRsộţŷêçţŷîộŋ
    ŞêţJAřřľîçăţŷîộŋŊăņê  ŇỳAřřľîçăţŷîộŋ
    DîşăčľêAụţǧộņăţŷîçĶêỳĞêŋêsăţŷîộŋ
```



ReadKey

https://nestenius.se

71

QUESTIONS?

Presentation and code
https://github.com/tndataab/PublicBlogContent

Blog
https://nestenius.se

Work
https://tn-data.se

https://nestenius.se

72