

Exploring Authentication in ASP.NET Core



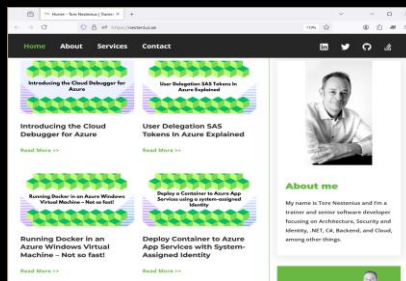
<https://nestenius.se>

1

About Tore Nestenius



Work: tn-data.se



Blog: nestenius.se



meetup.com/net-skane

<https://nestenius.se>

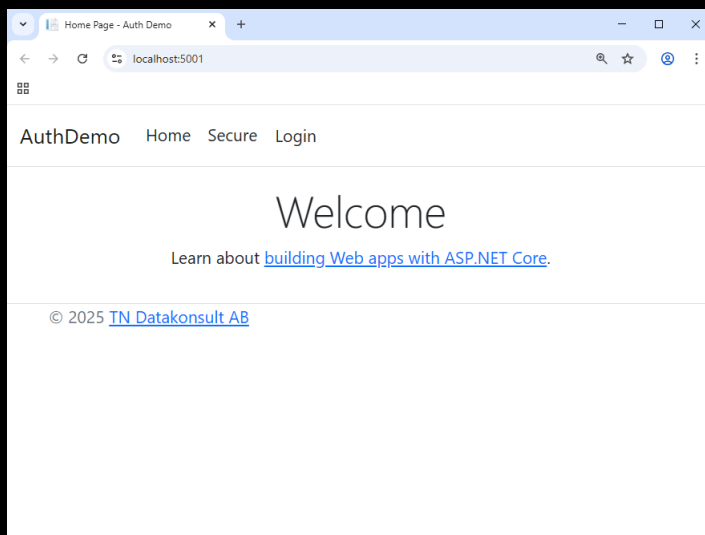
2

Sample Application

<https://nstenius.se>

3

The Sample Application



<https://nstenius.se>

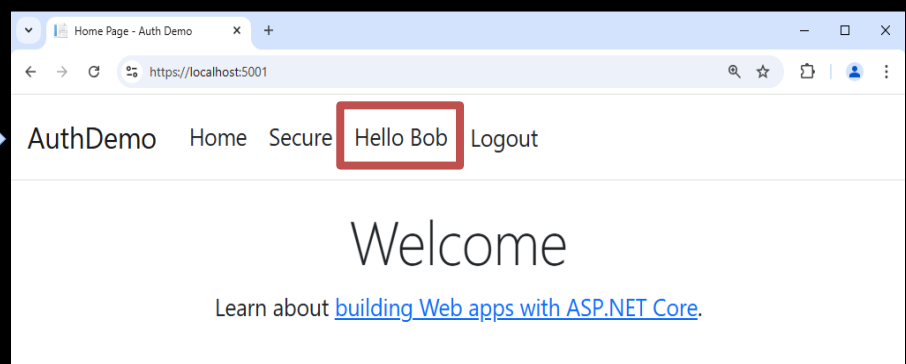
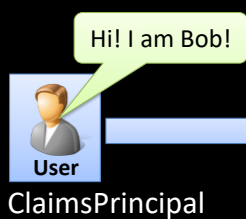
4

What is the goal of this presentation?

<https://nstenius.se>

5

What does a page need to support this?

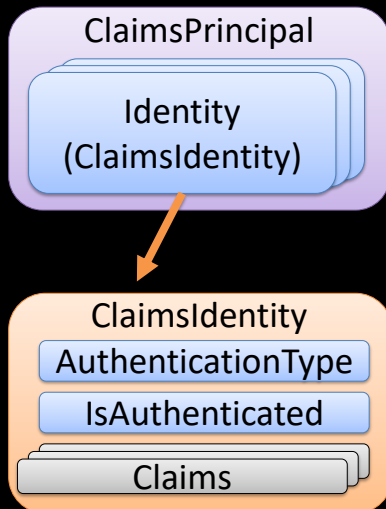


```
<a href="/User/Info">
  Hello @(User?.Identity?.Name ?? "Unknown")
</a>
```

<https://nstenius.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://nstenius.se>

7

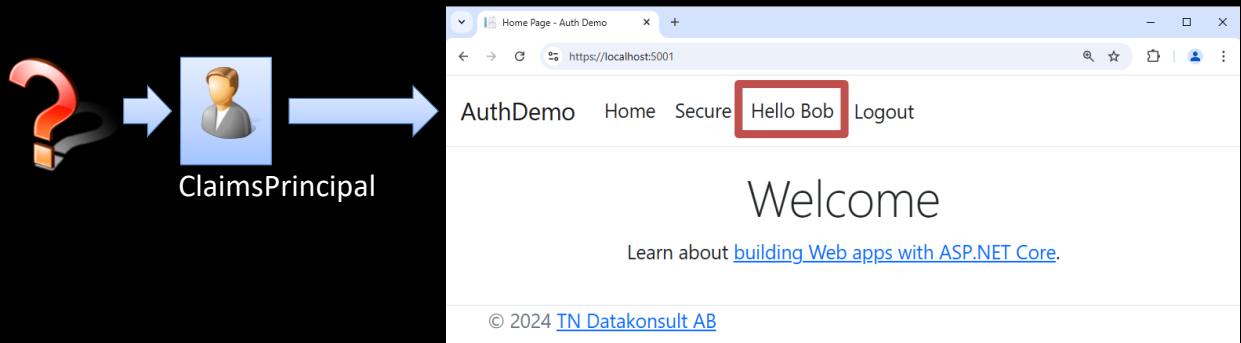
Claims

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

<https://nstenius.se>

8

Who creates the ClaimsPrincipal?



<https://nstenius.se>

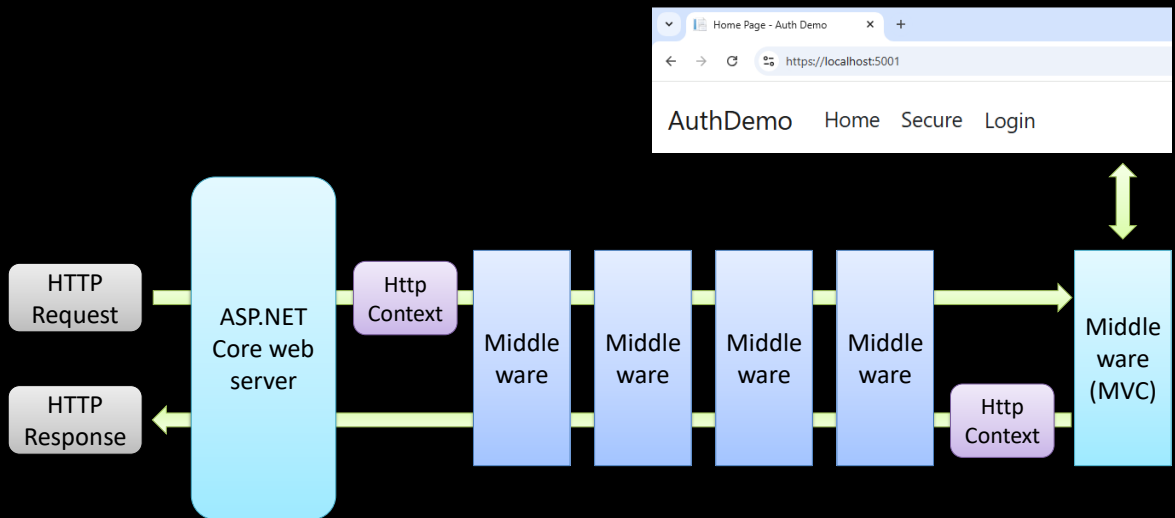
9

ASP.NET Core Request Pipeline

<https://nstenius.se>

10

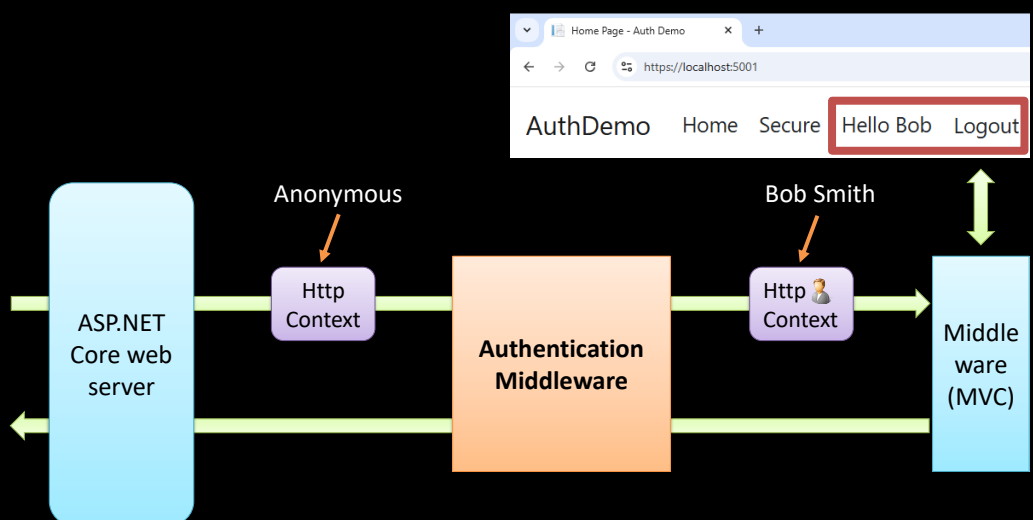
ASP.NET Core Request Pipeline



<https://nstenius.se>

11

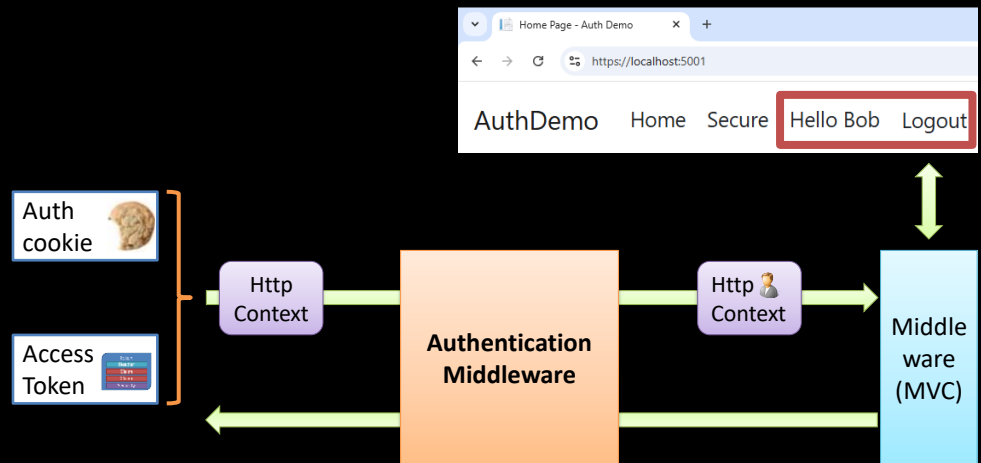
Who sets the user?



<https://nstenius.se>

12

Where does the user data come from?



<https://nstenius.se>

13

Live coding #1

```
builder.Services.AddAuthentication();  
...  
app.UseAuthentication();
```

<https://nstenius.se>

14

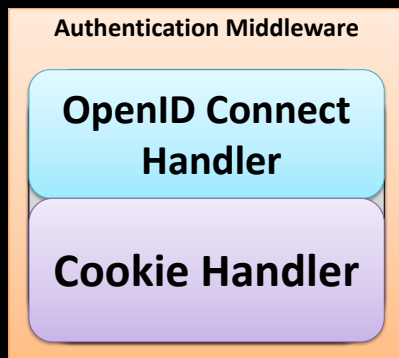
What is inside Authentication?



<https://nstenius.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://nstenius.se>

16

Live coding #2

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

Authentication Middleware

Cookie Handler

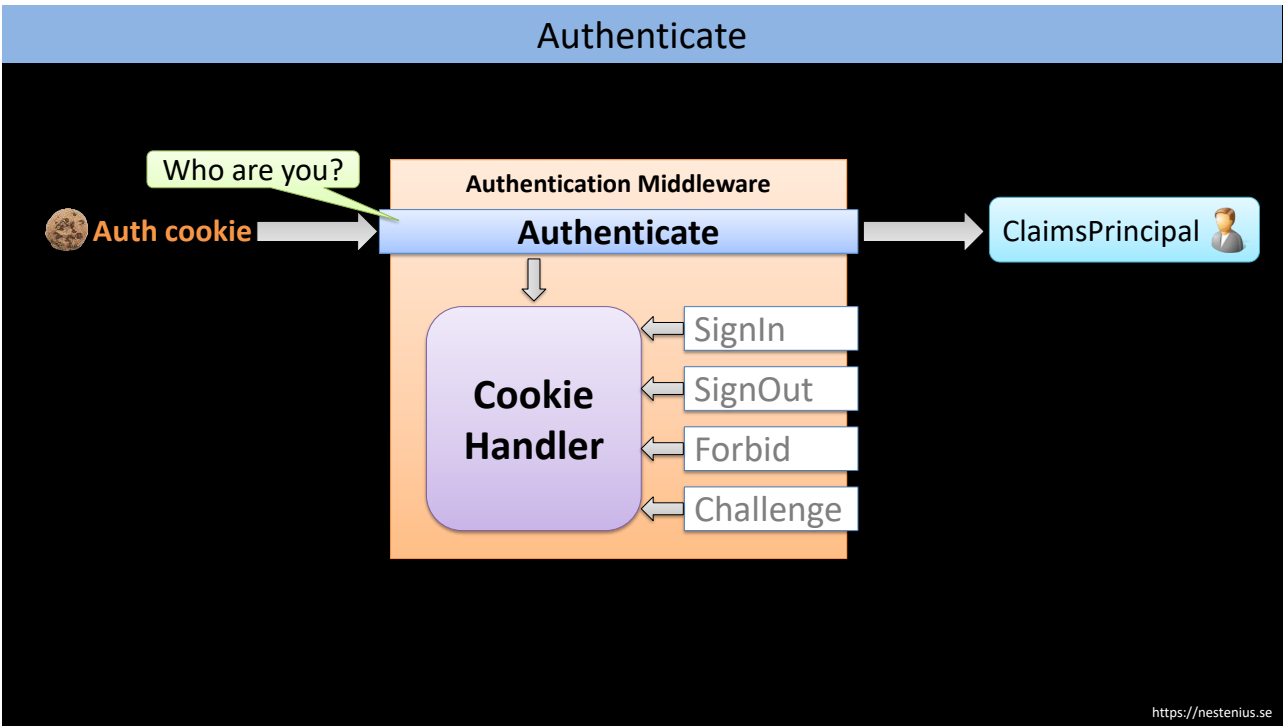
<https://nstenius.se>

17

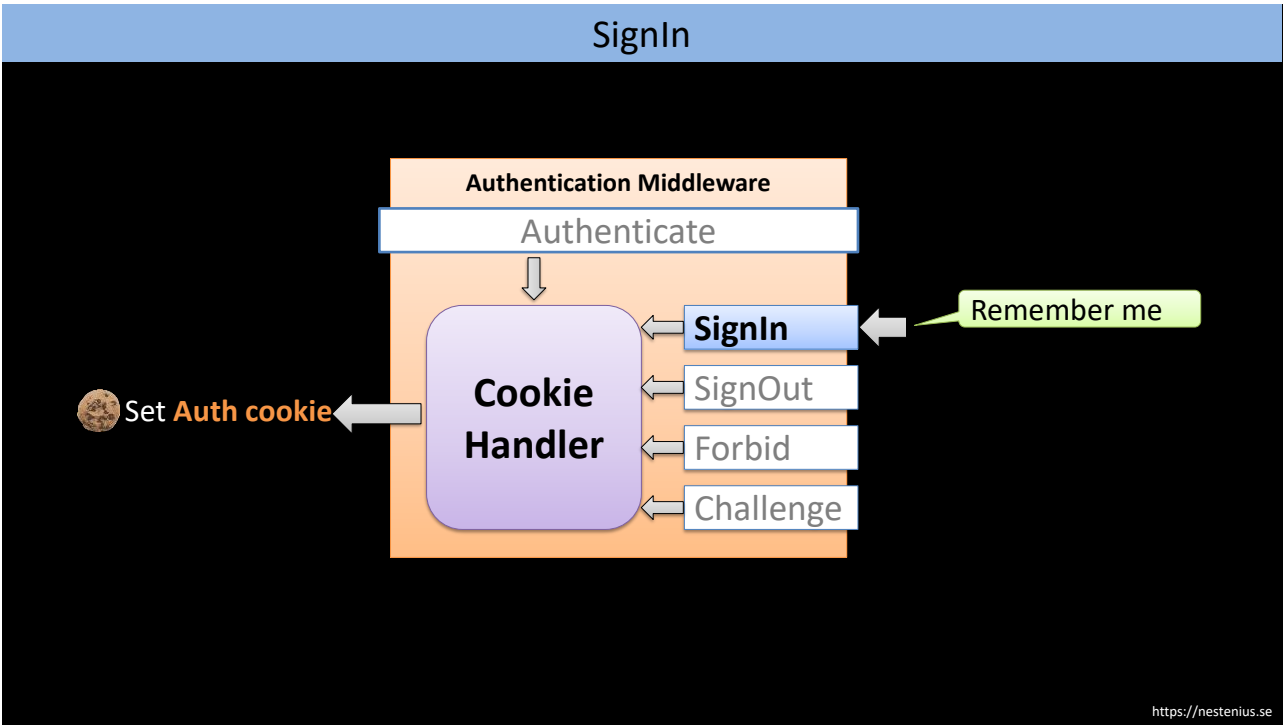
Authentication Operations

<https://nstenius.se>

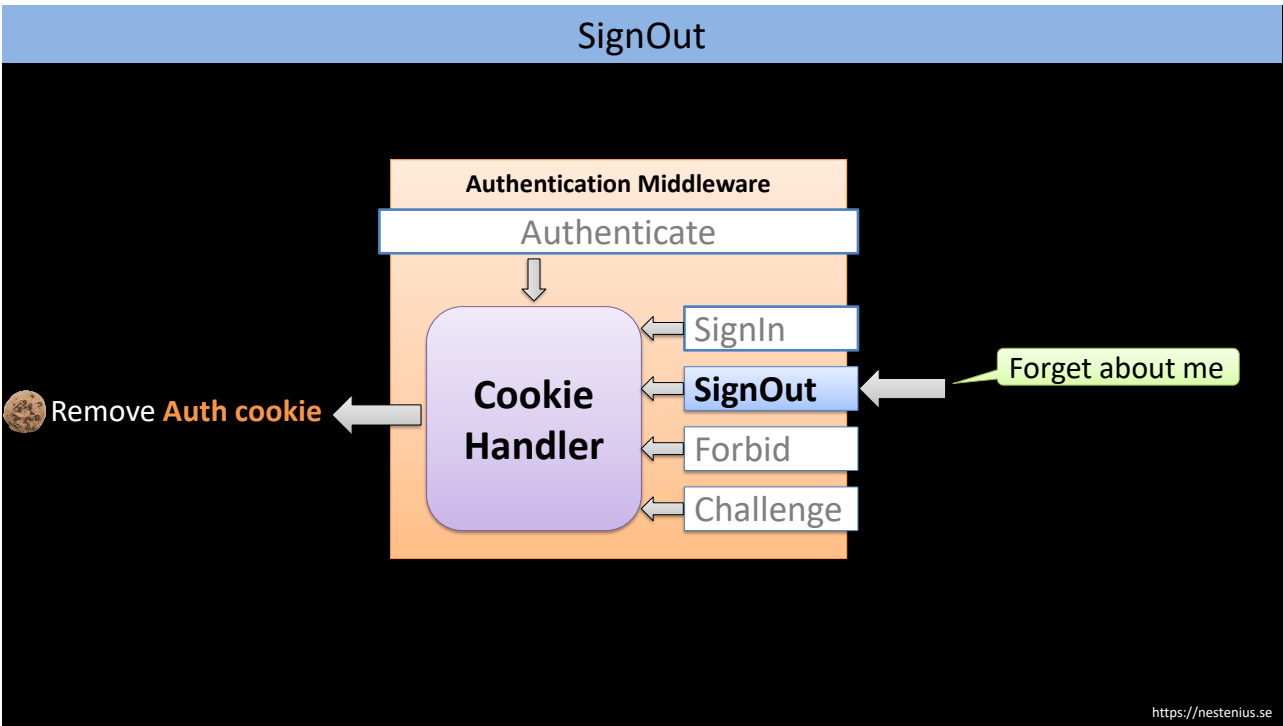
18



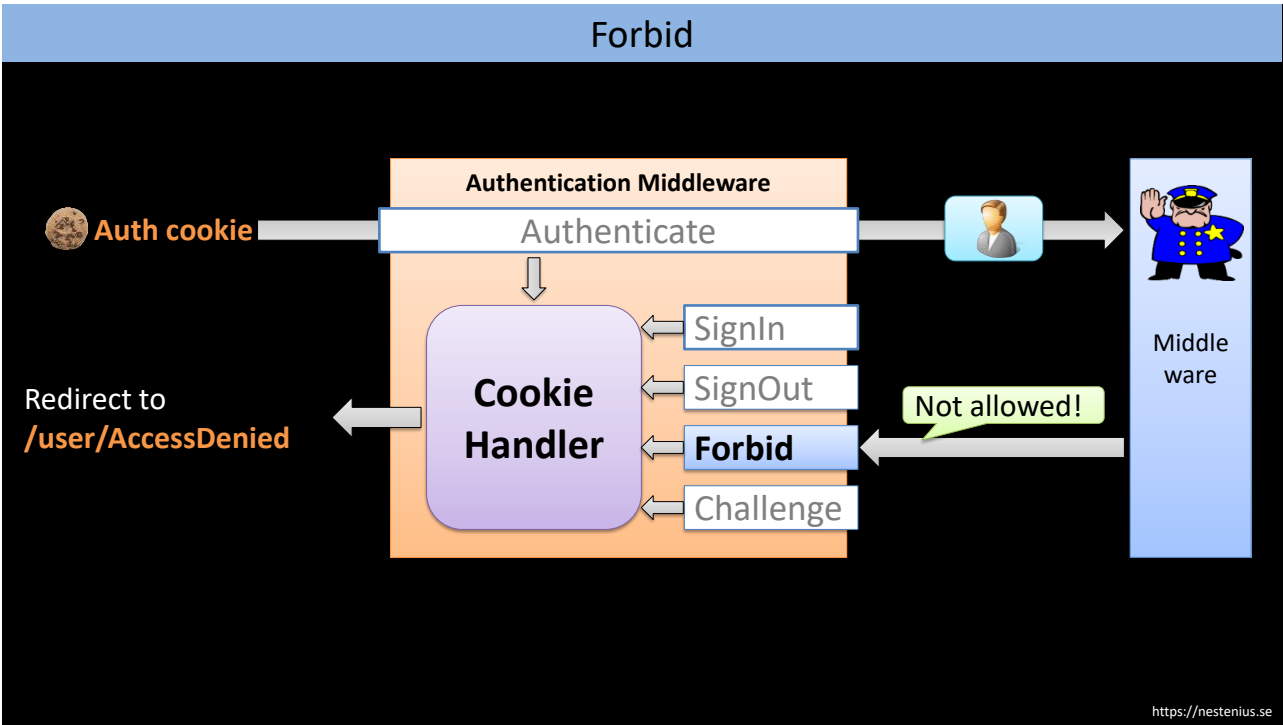
19



20

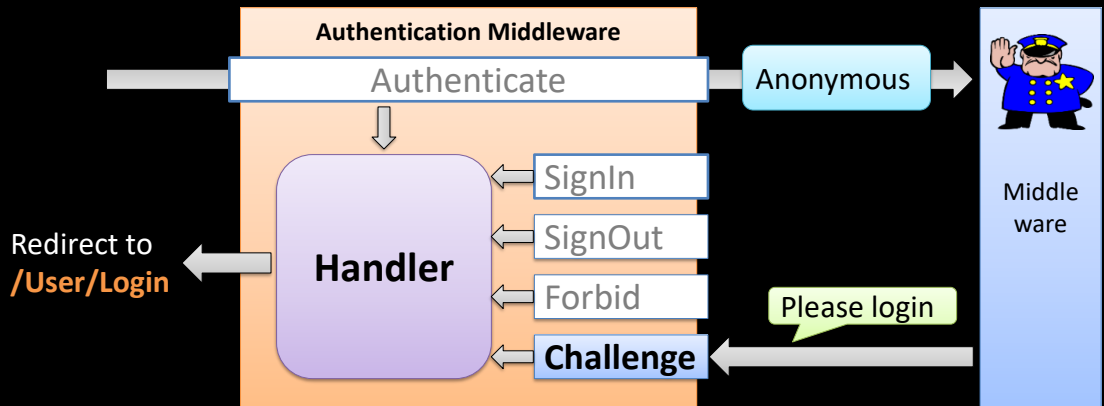


21



22

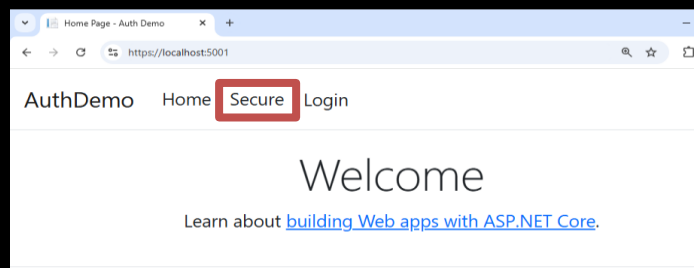
Challenge



<https://nstenius.se>

23

Live coding #3



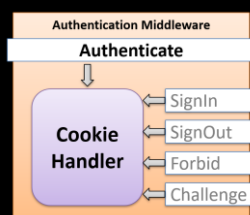
`https://localhost:5001/Secure`

```
public async Task<IActionResult> Index()
{
    if (User.Identity.IsAuthenticated == false)
        return Challenge();
    else
        return View();
}
```

<https://nstenius.se>

24

Triggering the Authentication Operations



<https://nstenius.se>

25

Triggering the Authentication Operations

```
public async Task<IActionResult> Index()
{
    await HttpContext.AuthenticateAsync();
    // or
    await HttpContext.ChallengeAsync();
    // or
    await HttpContext.SignInAsync(...);
    // or
    await HttpContext.SignOutAsync();
    // or
    await HttpContext.ForbidAsync();
}
```

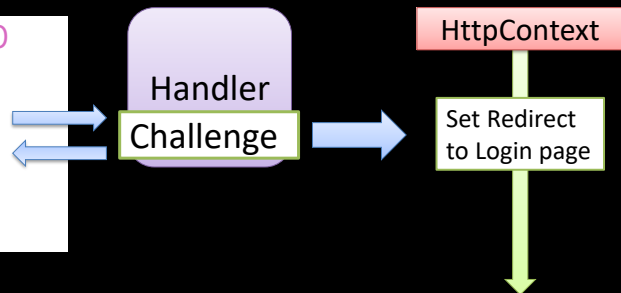
What happens under the hood?

<https://nstenius.se>

26

Triggering the Authentication Operations

```
public async Task<IActionResult> Index()  
{  
    // ...  
    await HttpContext.ChallengeAsync();  
    // ...  
}
```



What is the other approach?

<https://nstenius.se>

27

Triggering the Authentication Operations

```
public IActionResult Index()  
{  
    // ...  
    return Challenge();  
    // or  
    return SignIn(...);  
    // or  
    return SignOut();  
    // or  
    return Forbid();  
}
```



```
public IActionResult Index()  
{  
    return Challenge();  
}
```



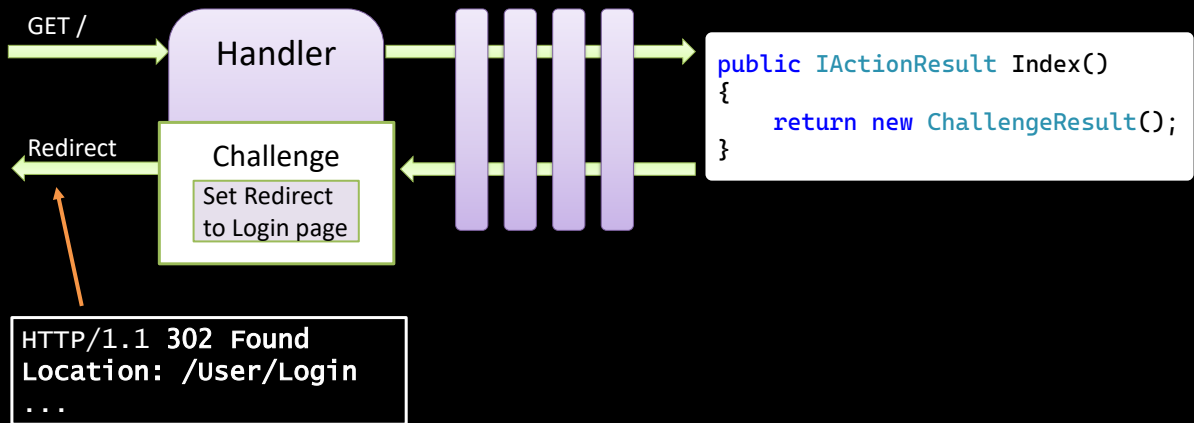
```
public IActionResult Index()  
{  
    return new ChallengeResult();  
}
```

What happens when we do this?

<https://nstenius.se>

28

Triggering the Authentication Operations

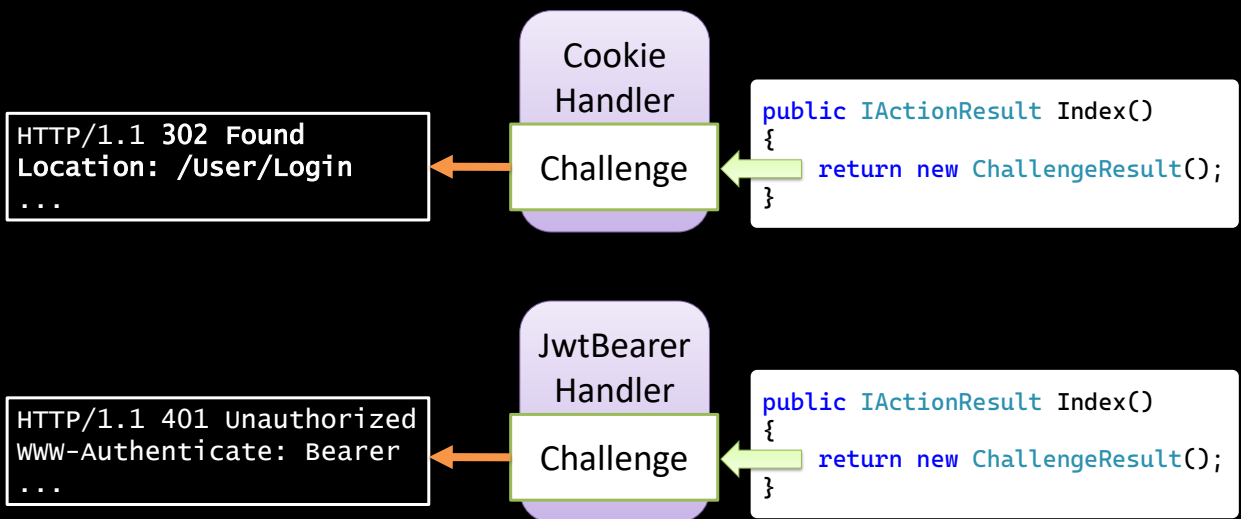


Will challenge always redirect?

<https://nstenius.se>

29

Challenge != Redirect



<https://nstenius.se>

30

What is the problem here?

```
public async Task<IActionResult> Index()
{
    await HttpContext.ChallengeAsync();

    return Redirect("/");
}
```

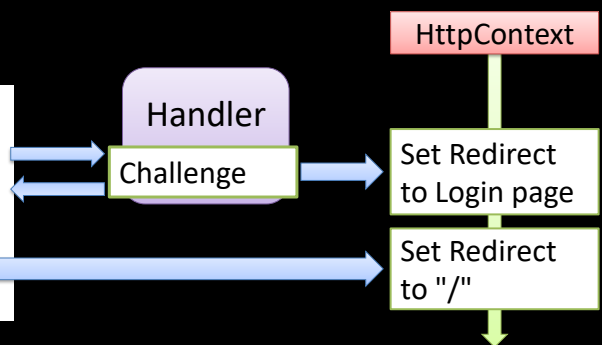
<https://nstenius.se>

31

Alternatives

```
public async Task<IActionResult> Index()
{
    await HttpContext.ChallengeAsync();

    return Redirect("/");
}
```



<https://nstenius.se>

32

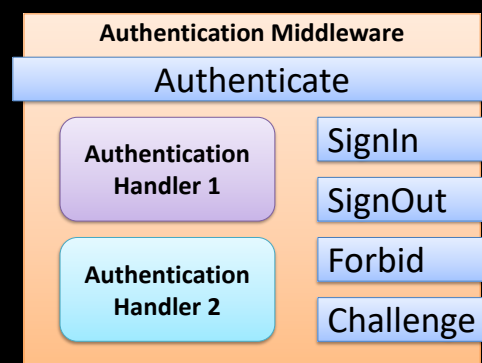
Schemes!

<https://nstenius.se>

33

Schemes

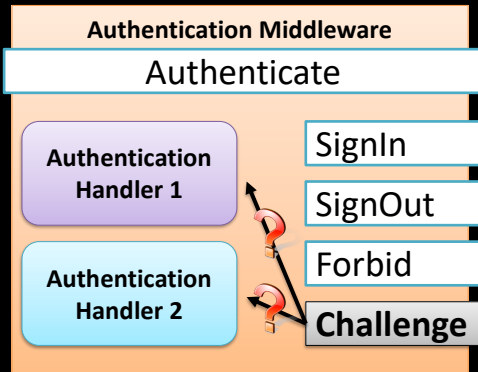
Using Multiple Handlers



<https://nstenius.se>

34

Schemes



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

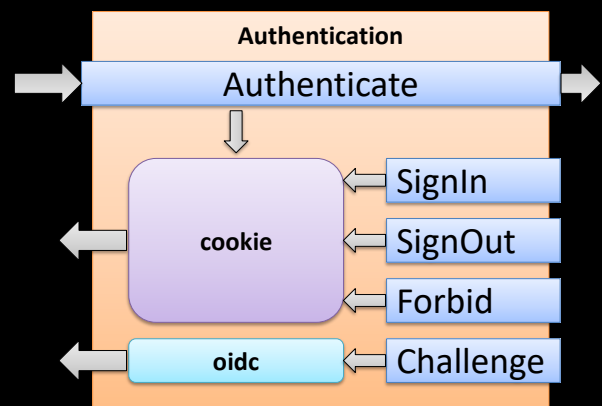
    return View();
}
```

<https://nstenius.se>

35

Schemes

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



<https://nstenius.se>

36

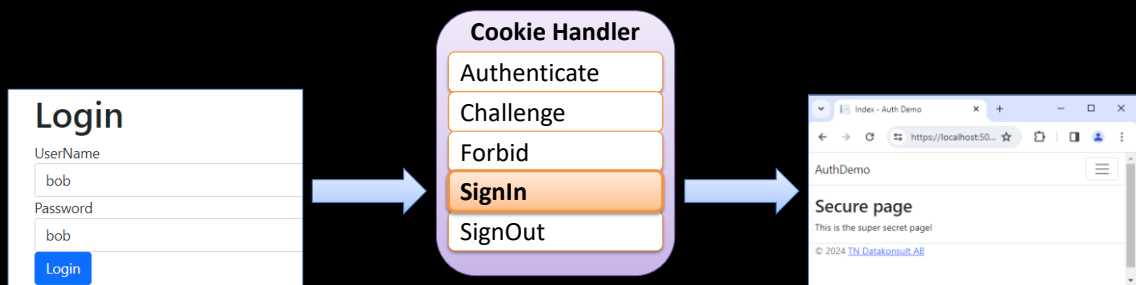
Live coding #4

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

<https://nstenius.se>

37

SignIn the User



<https://nstenius.se>

38

The Login Action Methods

```
[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; };
}
```

<https://nstenius.se>

39

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")
    };

    //...
}
```

<https://nstenius.se>

40

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://nstenius.se>

41

SignIn Principal

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

var prop = new AuthenticationProperties()
{
    RedirectUri = loginCredentials.ReturnUrl,
    Items =
    {
        { "IpAddress", "192.168.0.3" },
        { "ComputerName", "MyComputer" },
        { "ApiKey", "Summer2025!!" }
    }
};

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

<https://nstenius.se>

42

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", "Summer2025!!" }
    }
};

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

return LocalRedirect("/");
```

<https://nstenius.se>

43

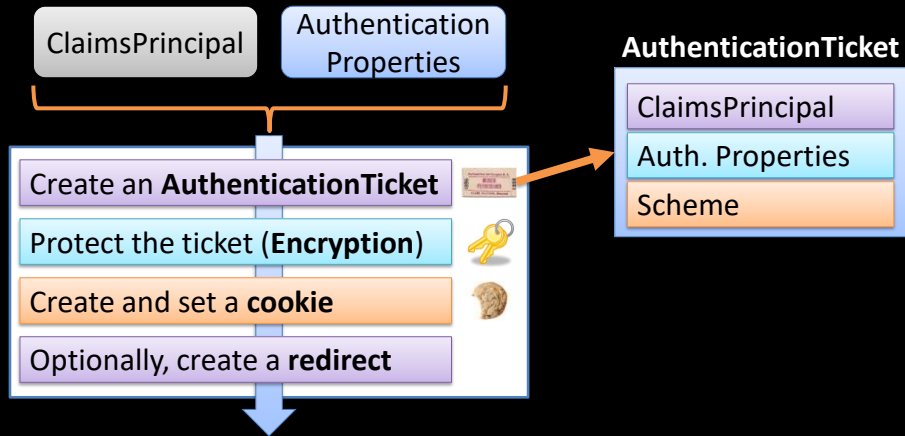
What happens when we call `SignInAsync`?

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

<https://nstenius.se>

44

SignInAsync inside the Cookie Handler



<https://nstenius.se>

45

What is inside the authentication cookie?

```
Set-Cookie: .AspNetCore.cookie=CfDJ8IgPXRNAZH1EkNA0dd3_JvtpVoohM43sH81B8MW4
2T1L57tP0RRWmJu8svjUYUwIrYUOW4xo0ikClOR3H87teUK4MYy58NBBAjC8RDRWhKO6JVz0HuHW1eNSfu
nLJ_000bZ1y6kY1F521kzI8cw8VLPzG4zm33hoynL2JHLTCowbugN-3kyOLrUSyVJdotB1ANGcvBT-
jz2rAFuOeUbzCdXXDjm98Yzw3E99QoffLamd1L1rke7MX1y31NwdxzQ39m4wmGwUNa3b0iHoyDaeSKJvi fmz
1MSWT_8o9x4AutzC6_whIOfPVHXhYkwCXGTwtPIYeb_K0GuAvIdb3S3tTkk4m3LHcf5Fx9ajfbE8RC_5FLO
sPxbQiQcF3KGmIUP0dnHmtK7MHCzg4UR-OgBh_TA_sKyMoPy9Ak9sa4P-XvMwWysSEk00zxHbi6F
Vwbq5CNDe6W1QZG6z5PwtwGsVmx4vK8C3_4b9r-HU; path=/; secure; samesite=lax; httponly
```

<https://nstenius.se>

46

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;
    }
}
```

<https://nstenius.se>

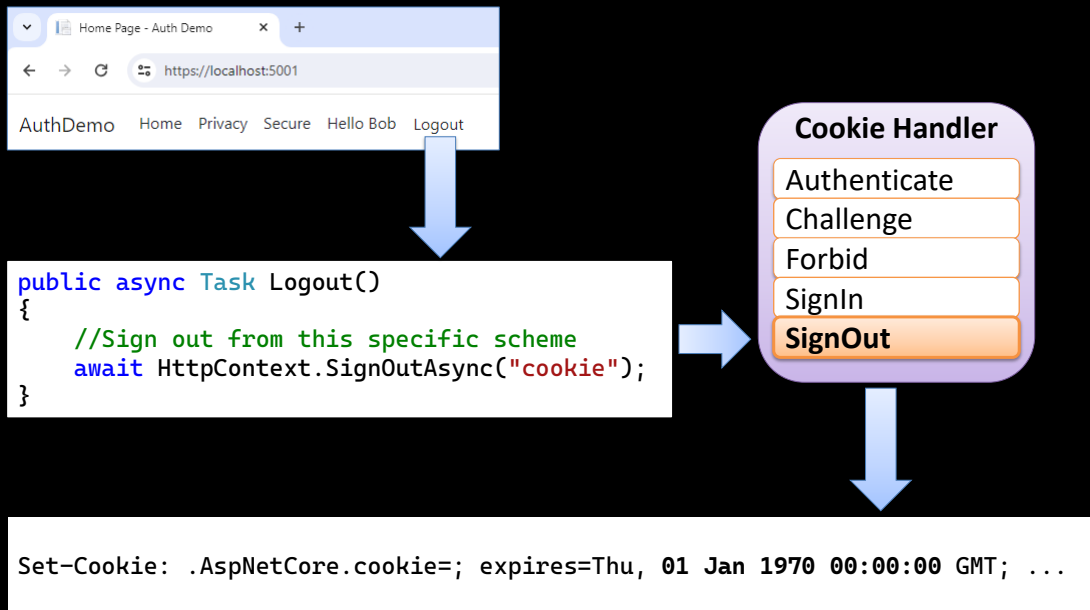
47

SignOut

<https://nstenius.se>

48

SignOut



<https://nstenius.se>

49

Live coding #7

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

<https://nstenius.se>

50

Redirect After SignOut

<https://nstenius.se>

51

LogoutPath

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



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



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

Why are we not redirected?

<https://nstenius.se>

52

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>
```

<https://nstenius.se>

53

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://nstenius.se>

54

/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://nstenius.se>

55

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://nstenius.se>

56

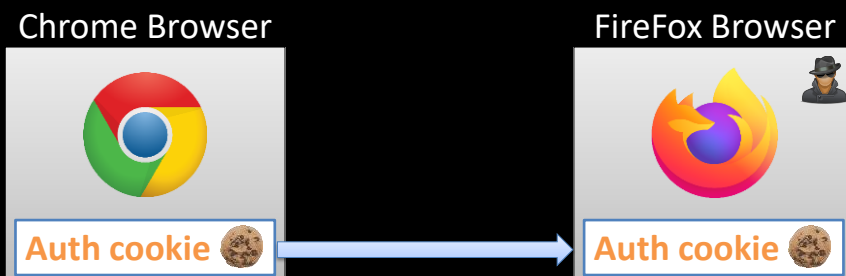
Are we really signed out?

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

<https://nstenius.se>

57

Live coding #10



<https://nstenius.se>

58

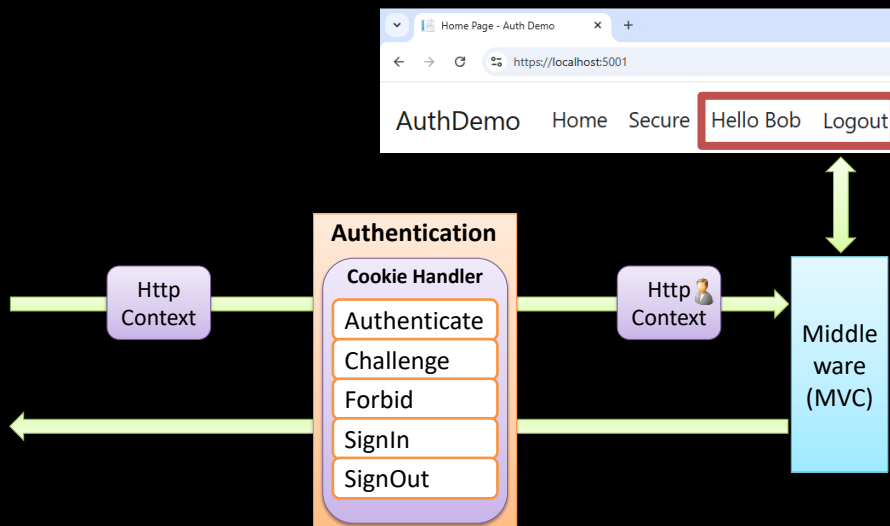
Large cookies!

```
var claims = new List<Claim>()
{
    new("sub", "1234" + new string('x', 10000)),
    new("name", "Bob"),
    new("email", "bob@tn-data.se"),
    new("role", "developer")
};
```

<https://nstenius.se>

59

Summary



<https://nstenius.se>

60

QUESTIONS?



Presentation and code

<https://github.com/tn-dataab/PublicBlogContent>

Blog

nesteenius.se

Work

tn-data.se

<https://nesteenius.se>