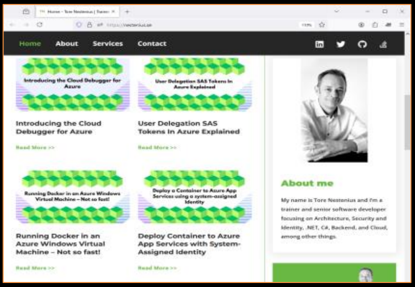
Top Authentication Fails in ASP.NET Core (and How to Avoid Them)

nestenius.se

MVP Microsoft® Most Valuable Professional

About Tore Nestenius

Work https://tn-data.se


Home
About
Services
Contact
Introducing the Cloud Debugger for Azure
User Delegation SAS Tokens In Azure Explained
Running Docker in an Azure Windows Virtual Machine – Not so fast!
Deploy a Container to Azure App Services using a system-assigned Identity

Introducing the Cloud Debugger for Azure

Read More >>

User Delegation SAS Tokens In Azure Explained

Read More >>

Running Docker in an Azure Windows Virtual Machine – Not so fast!

Read More >>

Deploy Container to Azure App Services with System-Assigned Identity

Read More >>

About me

My name is Tore Nestenius and I'm a trainer and senior software developer focusing on Architecture, Security and Identity, .NET, C#, Backend, and Cloud, among other things.

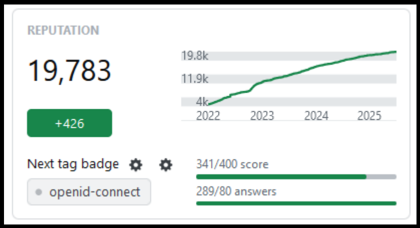
Blog https://nestenius.se

.NET Skåne

https://meetup.com/net-skane

Background

Stack Overflow



REPUTATION

19,783

+426

Next tag badge

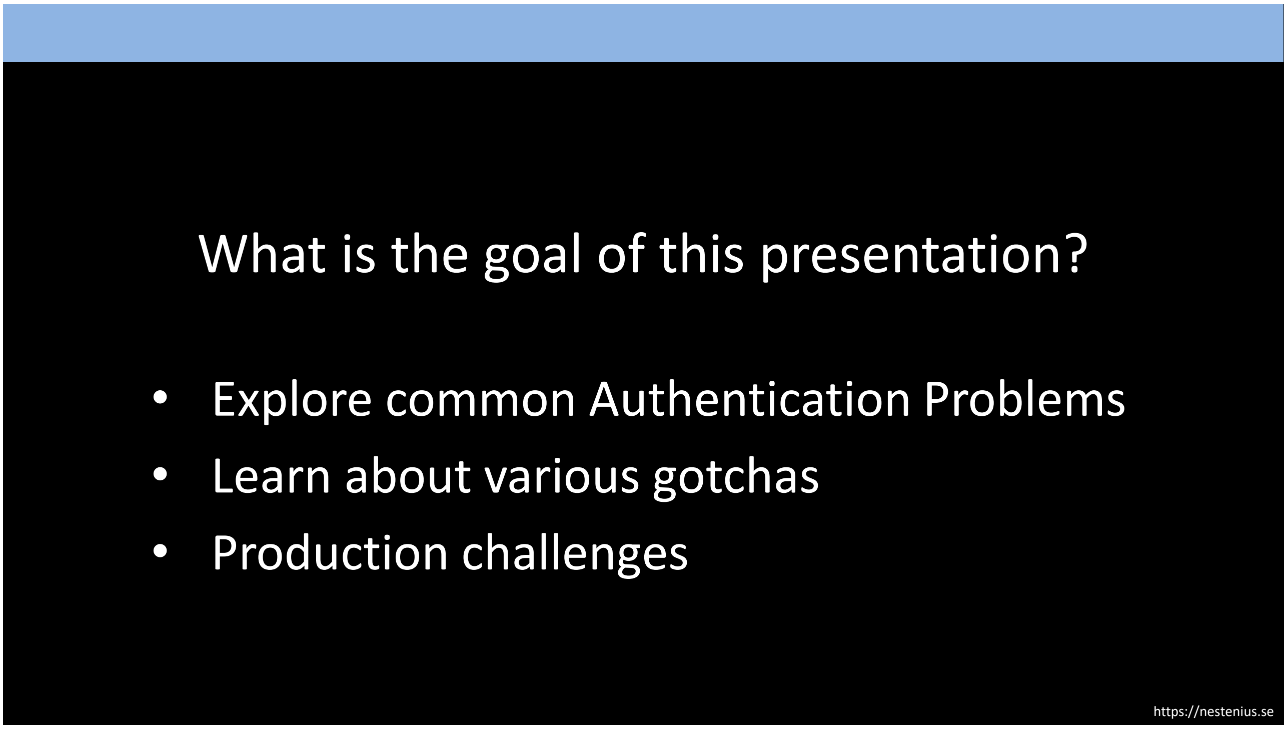
openid-connect

341/400 score

289/80 answers

Badges

- IdentityServer4
- OAuth 2.0
- OpenID-Connect
- Authentication
- JWT
- .NET
- C#
- ASP.NET Core



What is the goal of this presentation?

- Explore common Authentication Problems
- Learn about various gotchas
- Production challenges

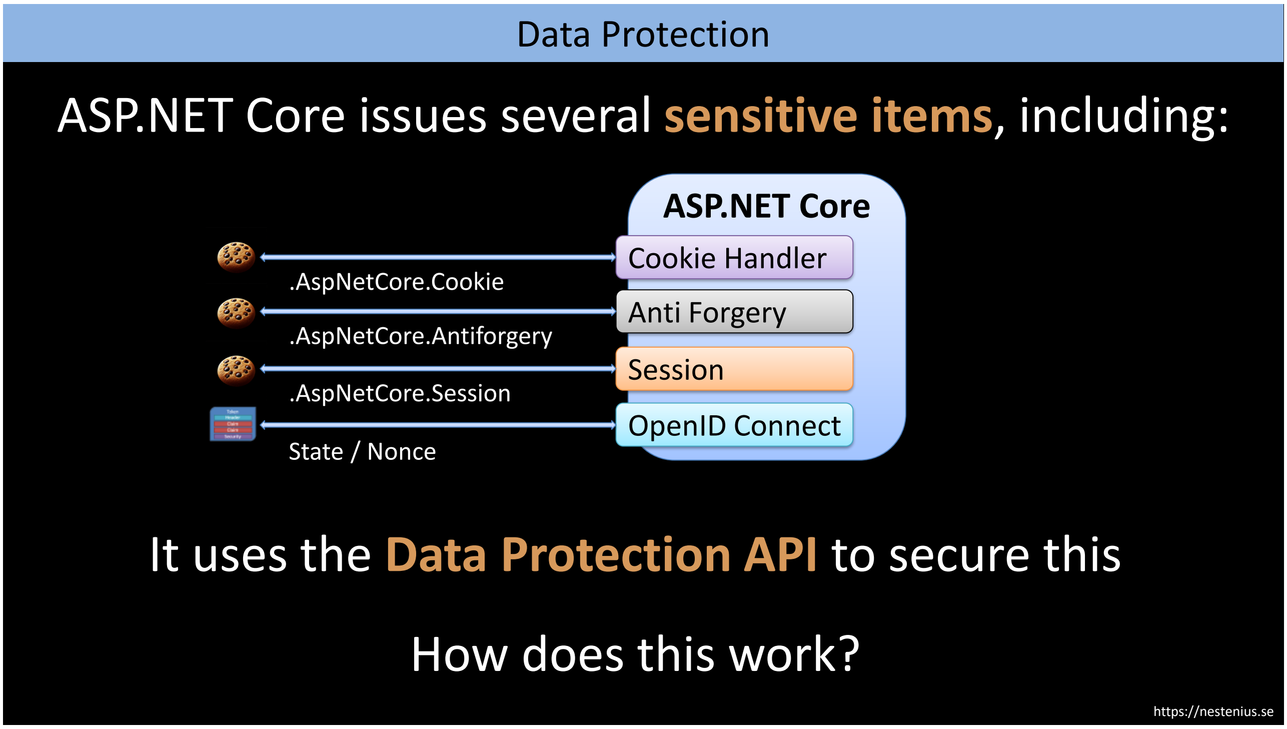
https://nestenius.se

5

Data Protection

https://nestenius.se

6



Data Protection

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

ASP.NET Core

Cookie Handler — .AspNetCore.Cookie

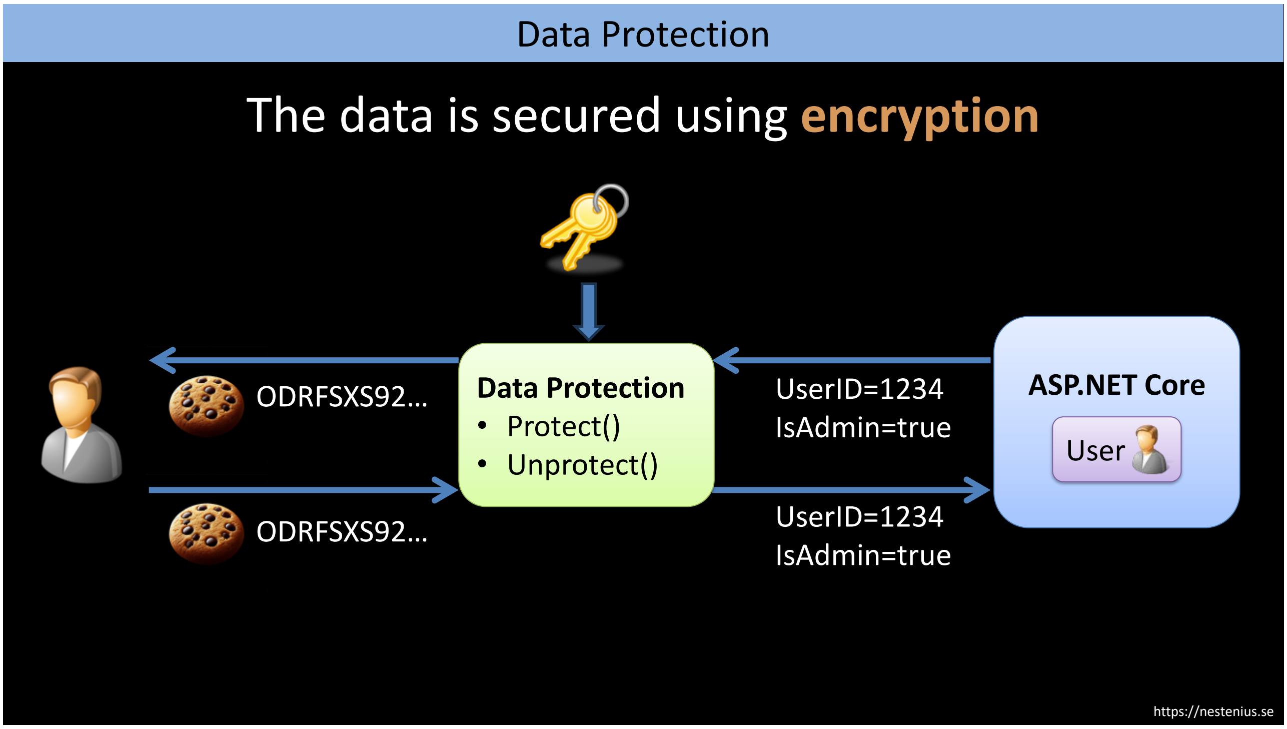
Anti Forgery — .AspNetCore.Antiforgery

Session — .AspNetCore.Session

OpenID Connect — State / Nonce

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

How does this work?



Data Protection

The data is secured using **encryption**

ODRFSXS92…

Data Protection

- Protect()
- Unprotect()

UserID=1234
IsAdmin=true

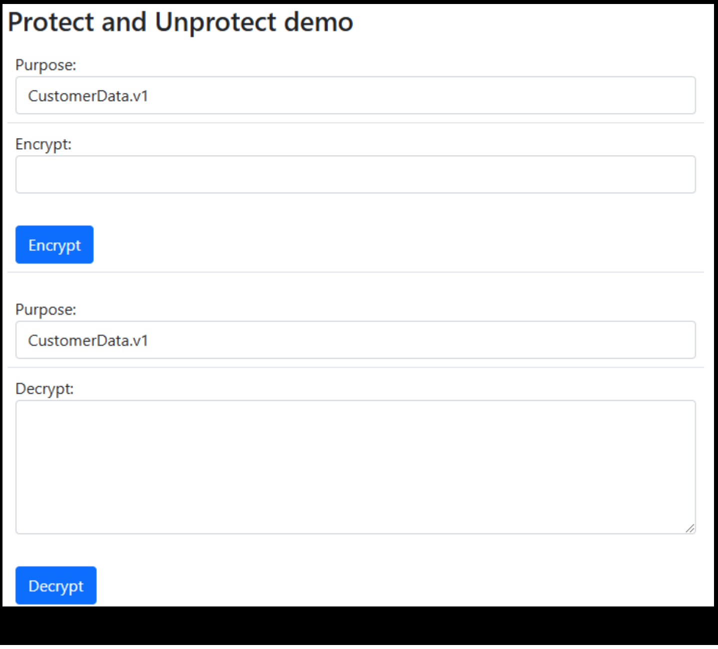
ASP.NET Core

User

ODRFSXS92…

UserID=1234
IsAdmin=true

Demonstration



Protect and Unprotect demo

Purpose: CustomerData.v1

Encrypt:

Purpose: CustomerData.v1

Decrypt:

https://nestenius.se

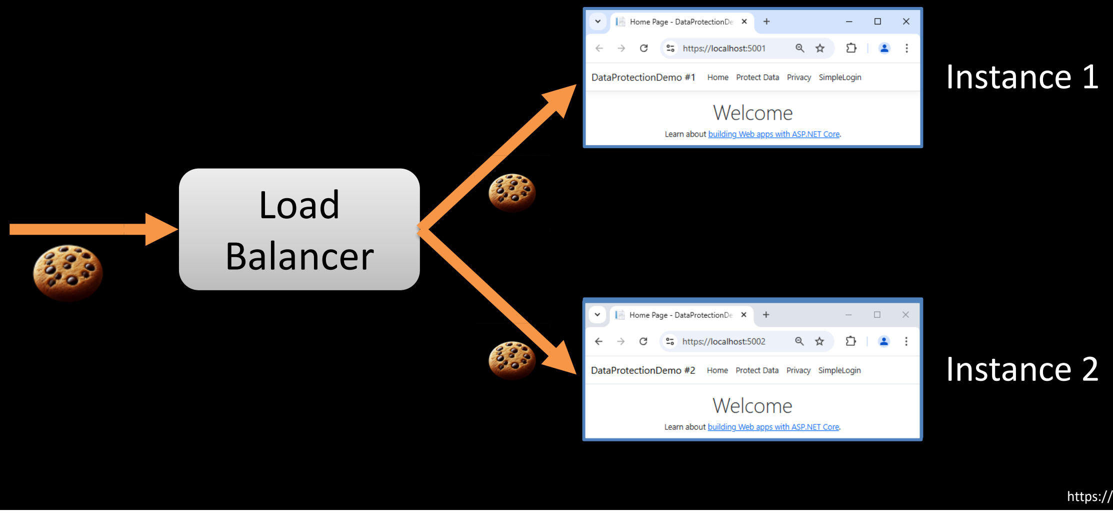
Data Protection in Production

```
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

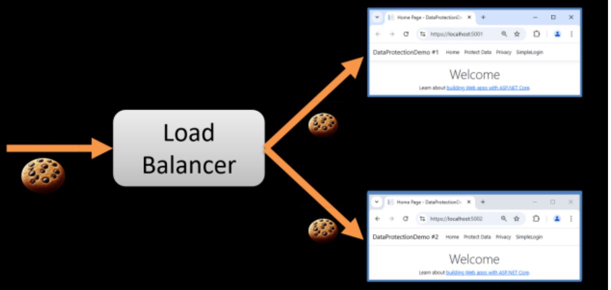
Example Architecture


Load Balancer
Home Page - DataProtectionDe
https://localhost:5001
DataProtectionDemo #1 Home Protect Data Privacy SimpleLogin
Welcome
Learn about building Web apps with ASP.NET Core.
Instance 1
Home Page - DataProtectionDe
https://localhost:5002
DataProtectionDemo #2 Home Protect Data Privacy SimpleLogin
Welcome
Learn about building Web apps with ASP.NET Core.
Instance 2

Data Protection

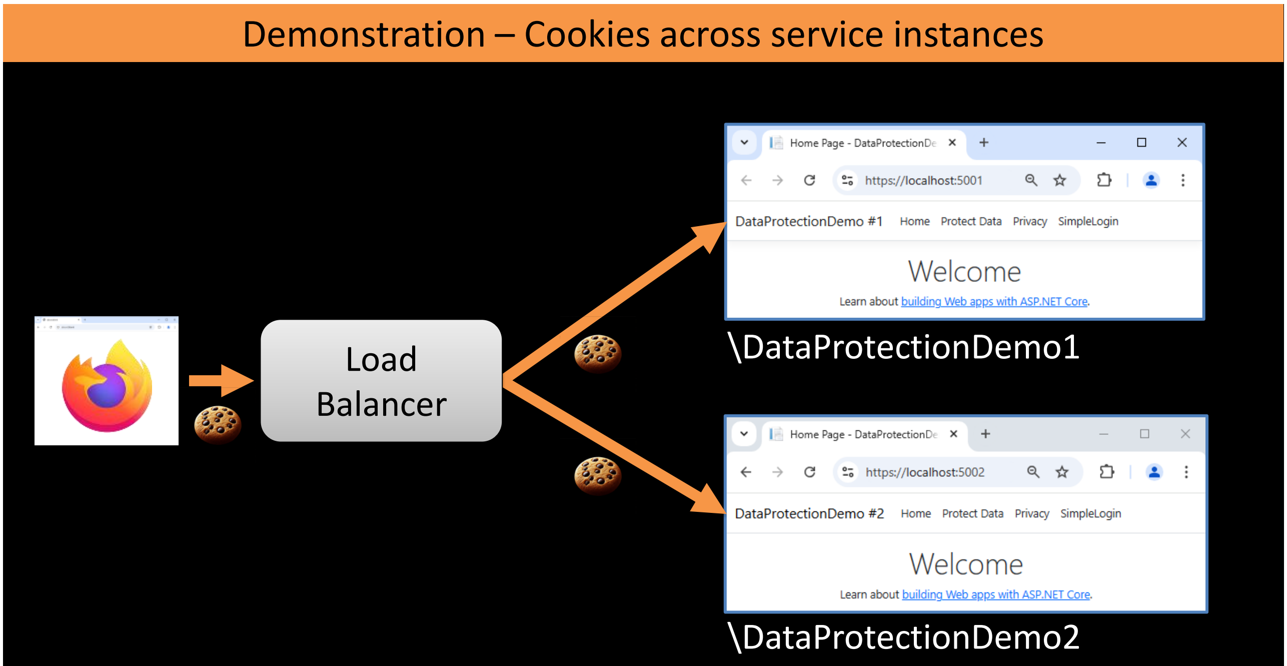
What can go wrong here?

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



What is the problem?

Demonstration – Cookies across service instances



Load Balancer

Home Page - DataProtectionDe

https://localhost:5001

DataProtectionDemo #1 Home Protect Data Privacy SimpleLogin

Welcome

Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo1

Home Page - DataProtectionDe

https://localhost:5002

DataProtectionDemo #2 Home Protect Data Privacy SimpleLogin

Welcome

Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo2

13

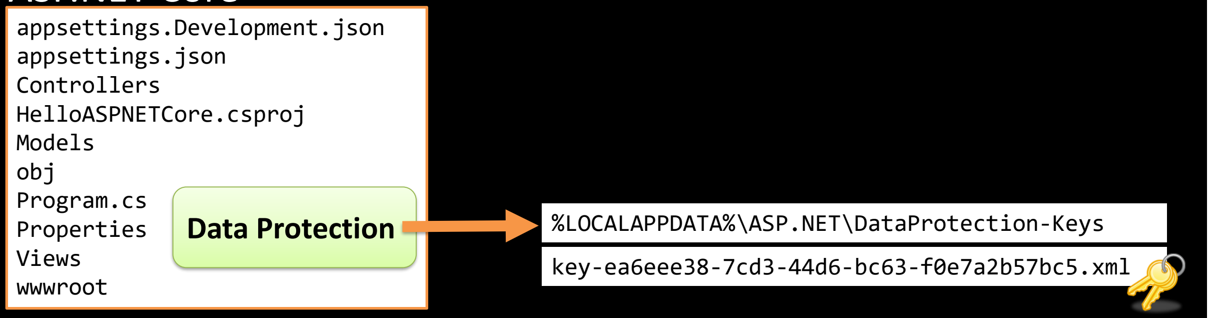
Data Protection at Startup

14

Data Protection

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

ASP.NET Core



```
appsettings.Development.json
appsettings.json
Controllers
HelloASPNETCore.csproj
Models
obj
Program.cs
Properties
Views
wwwroot
```

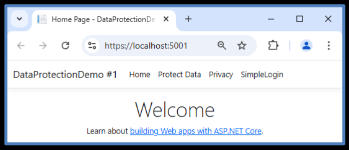
```
%LOCALAPPDATA%\ASP.NET\DataProtection-Keys
key-ea6eee38-7cd3-44d6-bc63-f0e7a2b57bc5.xml
```

```
>dotnet run
```

Data Protection

The following happens in our example:

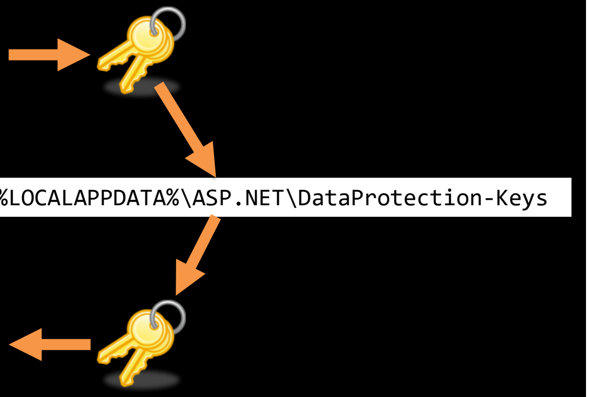
Instance #1



DataProtectionDemo #1 Home Protect Data Privacy SimpleLogin

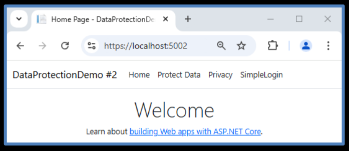
Welcome

Learn about building Web apps with ASP.NET Core.



```
%LOCALAPPDATA%\ASP.NET\DataProtection-Keys
```

Instance #2



DataProtectionDemo #2 Home Protect Data Privacy SimpleLogin

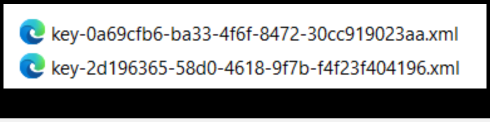
Welcome

Learn about building Web apps with ASP.NET Core.

What can go wrong here?

Data Protection

We can get a race condition!



key-0a69cfb6-ba33-4f6f-8472-30cc919023aa.xml
key-2d196365-58d0-4618-9f7b-f4f23f404196.xml

Instance #1

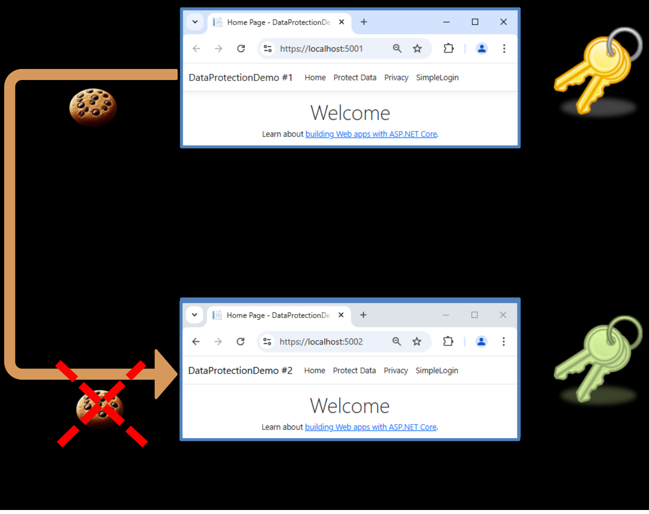
```
2025-02-03T16:28:44 Repository contains no viable default key.
                    Caller should generate a key with immediate activation.
2025-02-03T16:28:44 Policy resolution states that a new key should be added to the key ring.
2025-02-03T16:28:44 Creating key {0a69cfb6-ba33-4f6f-8472-30cc919023aa} with
                    creation date 2025-02-03 15:28:44Z, activation date 2025-02-03 15:28:44Z,
                    and expiration date 2025-05-04 15:28:44Z.
```

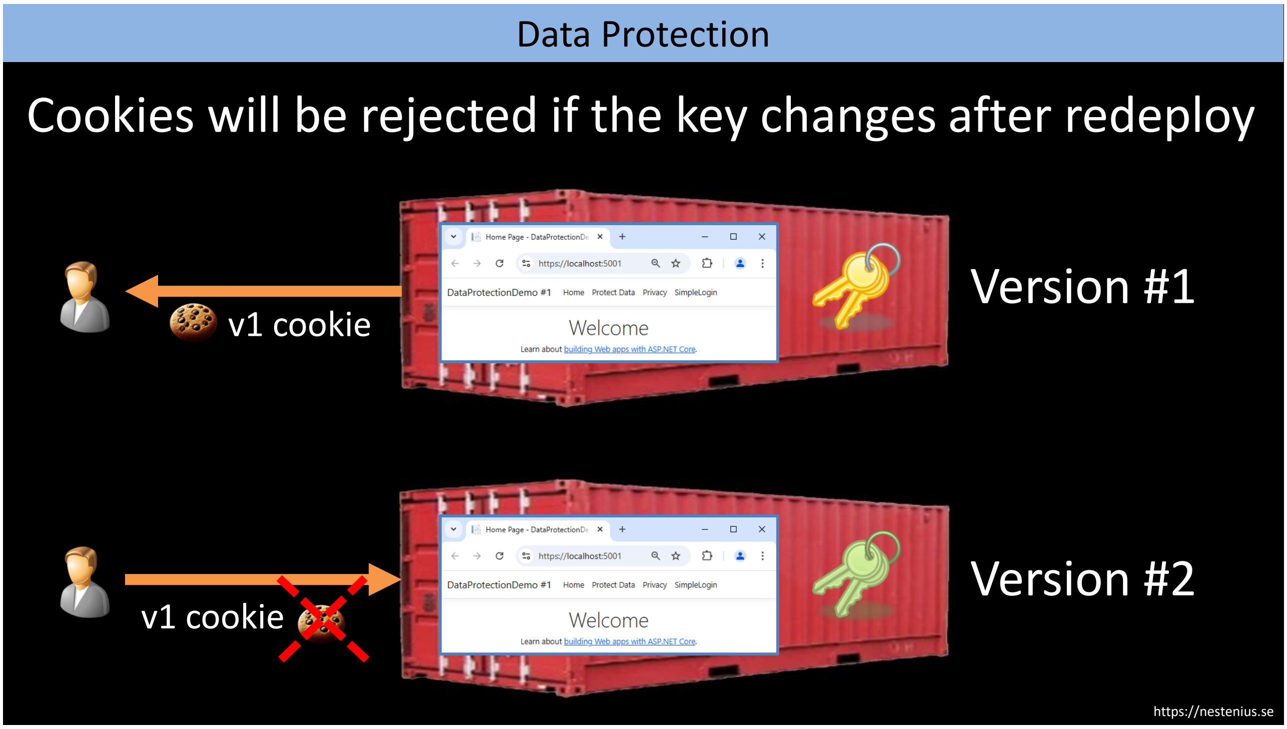
Instance #2

```
2025-02-03T16:28:44 Repository contains no viable default key.
                    Caller should generate a key with immediate activation.
2025-02-03T16:28:44 Policy resolution states that a new key should be added to the key ring.
2025-02-03T16:28:44 Creating key {2d196365-58d0-4618-9f7b-f4f23f404196} with
                    creation date 2025-02-03 15:28:44Z, activation date 2025-02-03 15:28:44Z,
                    and expiration date 2025-05-04 15:28:44Z.
```

Data Protection

Cookies will be rejected if the expected key is not found



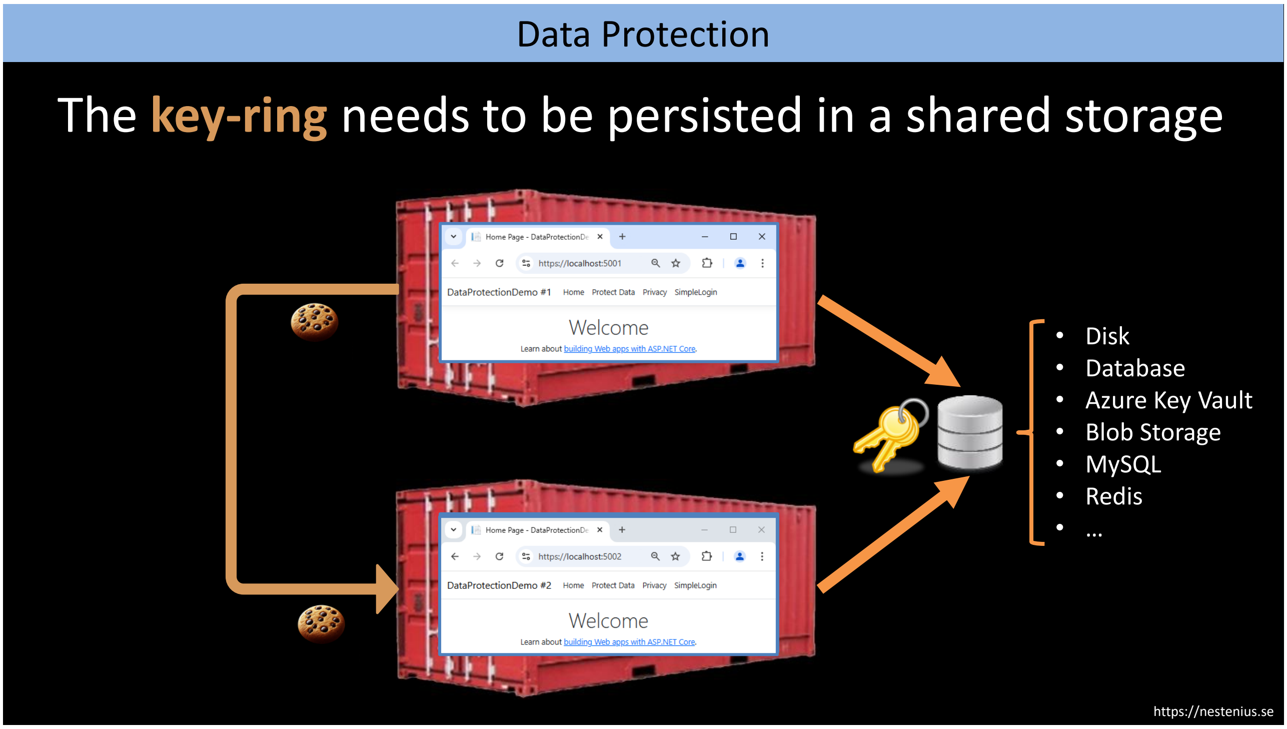


Data Protection

Cookies will be rejected if the key changes after redeploy

https://nestenius.se

19



Data Protection

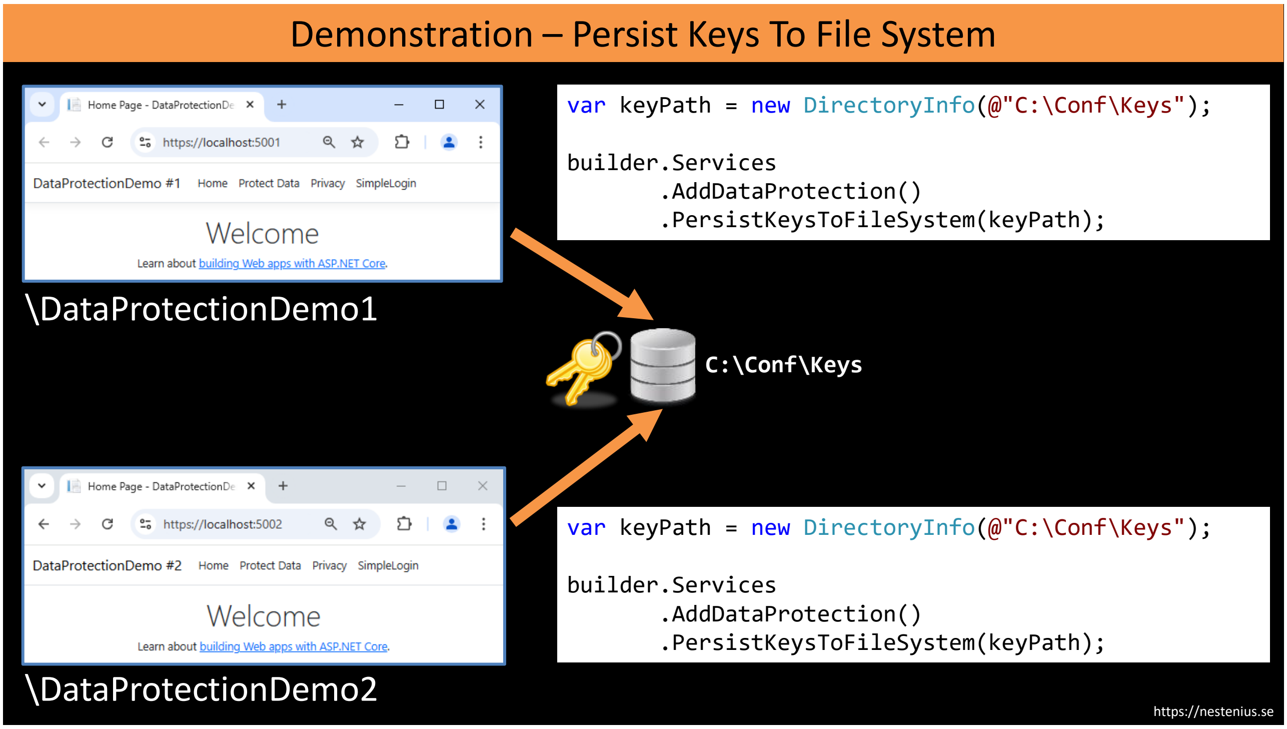
The **key-ring** needs to be persisted in a shared storage

- Disk
- Database
- Azure Key Vault
- Blob Storage
- MySQL
- Redis
- …

https://nestenius.se

20

Demonstration – Persist Keys To File System



Welcome

Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo1

```
var keyPath = new DirectoryInfo(@"C:\Conf\Keys");

builder.Services
       .AddDataProtection()
       .PersistKeysToFileSystem(keyPath);
```

C:\Conf\Keys

Welcome

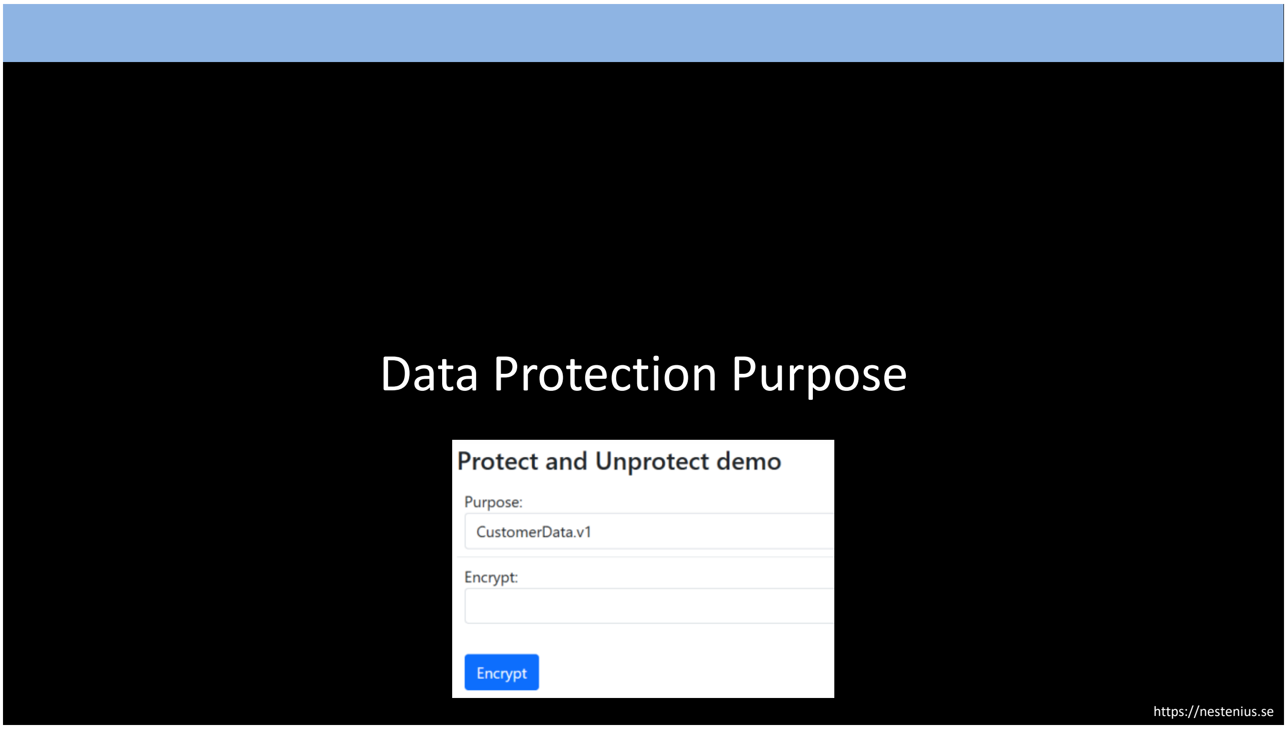
Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo2

```
var keyPath = new DirectoryInfo(@"C:\Conf\Keys");

builder.Services
       .AddDataProtection()
       .PersistKeysToFileSystem(keyPath);
```

https://nestenius.se

Data Protection Purpose



Protect and Unprotect demo

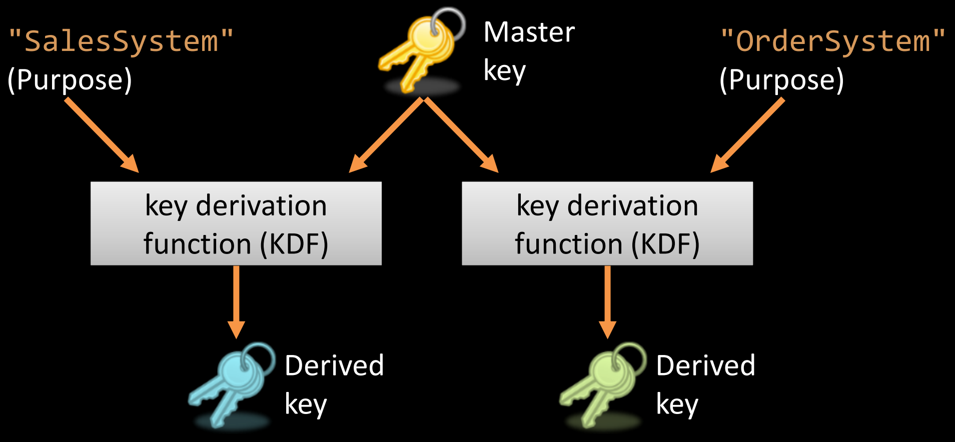
Purpose: CustomerData.v1

Encrypt:

https://nestenius.se

Data Protection Purpose

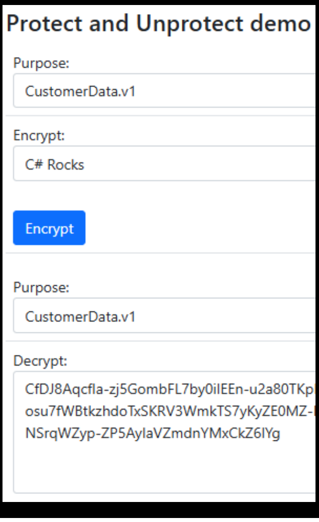
The **purpose** is used to create isolation



Demonstration - Purpose

Why does this not work?

Instance #1



Protect and Unprotect demo

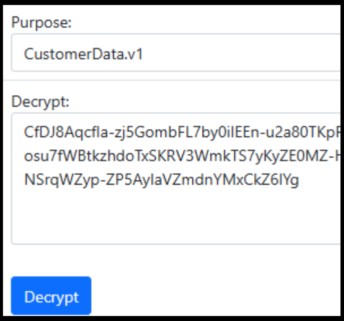
Purpose:
CustomerData.v1

Encrypt:
C# Rocks

Purpose:
CustomerData.v1

Decrypt:
CfDJ8Aqcfla-zj5GombFL7by0iIEEn-u2a80TKp
osu7fWBtkzhdoTxSKRV3WmkTS7yKyZE0MZ-
NSrqWZyp-ZP5AylaVZmdnYMxCkZ6lYg

Instance #2

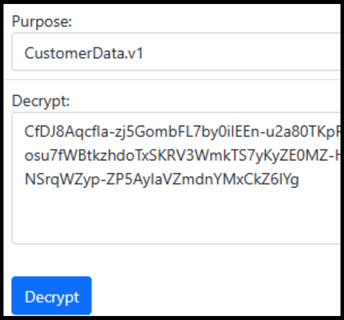


Purpose:
CustomerData.v1

Decrypt:
CfDJ8Aqcfla-zj5GombFL7by0iIEEn-u2a80TKpF
osu7fWBtkzhdoTxSKRV3WmkTS7yKyZE0MZ-H
NSrqWZyp-ZP5AylaVZmdnYMxCkZ6lYg

Data Protection Purpose

We actually have two purposes!



Purpose:

CustomerData.v1

Decrypt:

CfDJ8Aqcfla-zj5GombFL7by0ilEEn-u2a80TKpF
osu7fWBtkzhdoTxSKRV3WmkTS7yKyZE0MZ-H
NSrqWZyp-ZP5AylaVZmdnYMxCkZ6lYg

```
Performing unprotect operation to key {bc7e21f3-69e3-45ce-bb5b-e9007bac85ed}
with purposes ('C:\Conf\Demo - Data Protection\DataProtectionDemo1\',
               'CustomerData.v1').
```

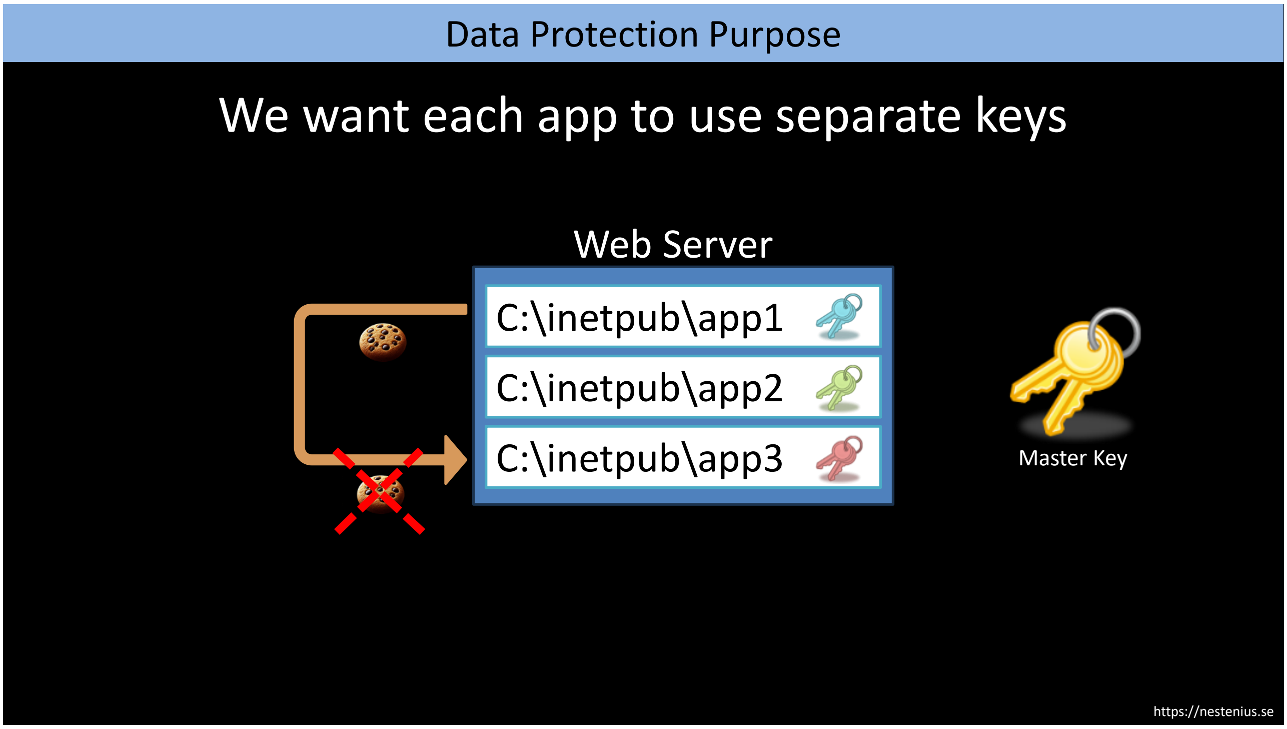
25

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').
```

26



Data Protection Purpose

We want each app to use separate keys

Web Server

C:\inetpub\app1

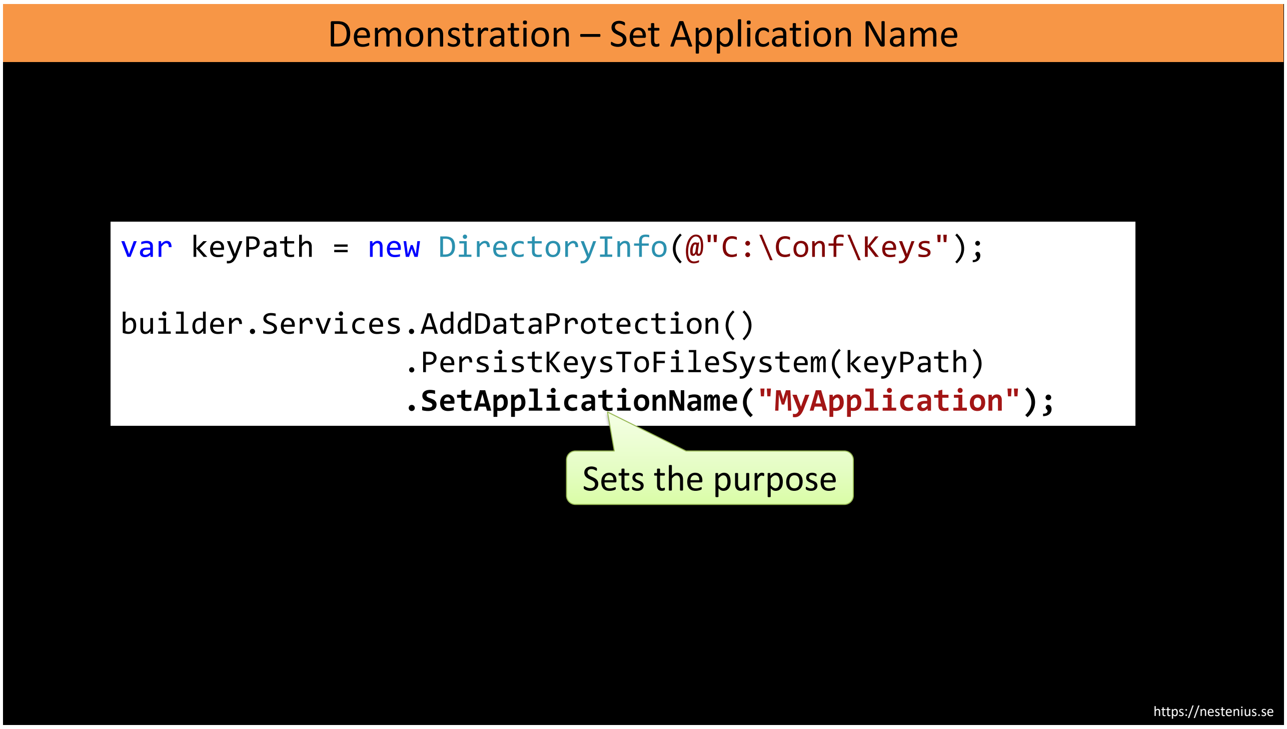
C:\inetpub\app2

C:\inetpub\app3

Master Key

https://nestenius.se

27



Demonstration – Set Application Name

```
var keyPath = new DirectoryInfo(@"C:\Conf\Keys");

builder.Services.AddDataProtection()
                    .PersistKeysToFileSystem(keyPath)
                    .SetApplicationName("MyApplication");
```

Sets the purpose

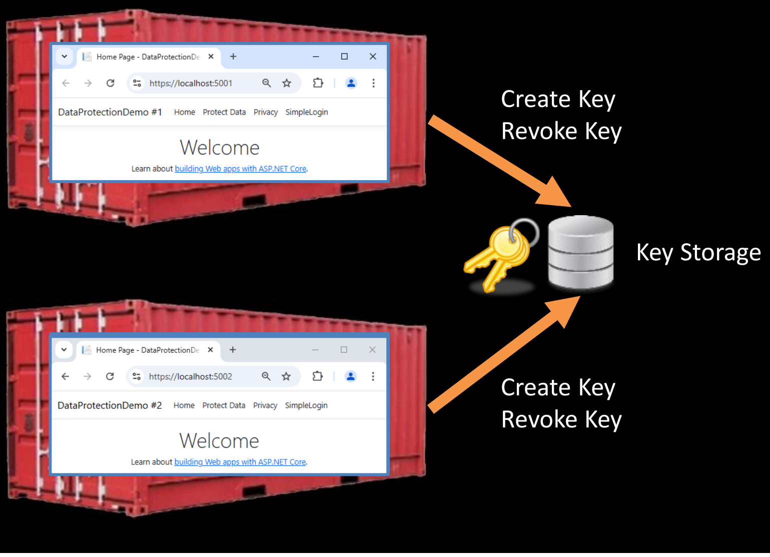
https://nestenius.se

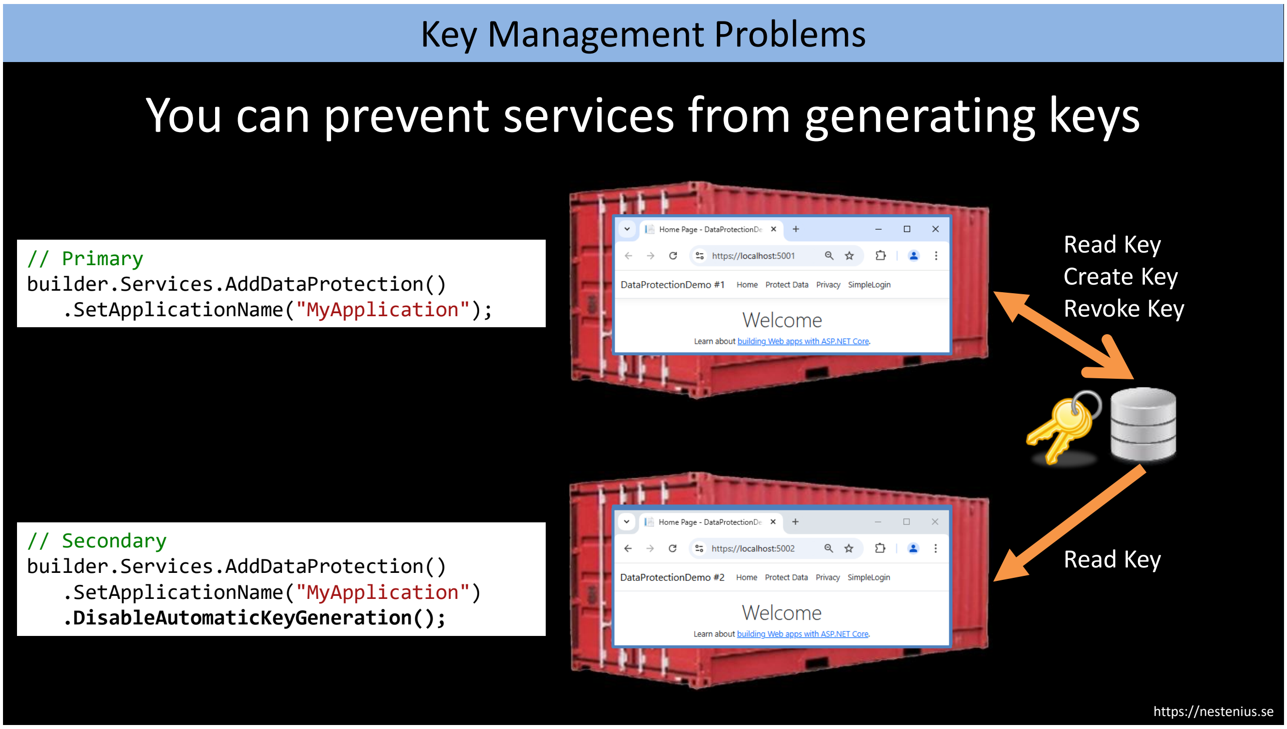
28

Key Management Problems

Key Management Problems

In this setup, both services wants to manage the keys





Key Management Problems

You can prevent services from generating keys

```
// Primary
builder.Services.AddDataProtection()
   .SetApplicationName("MyApplication");
```

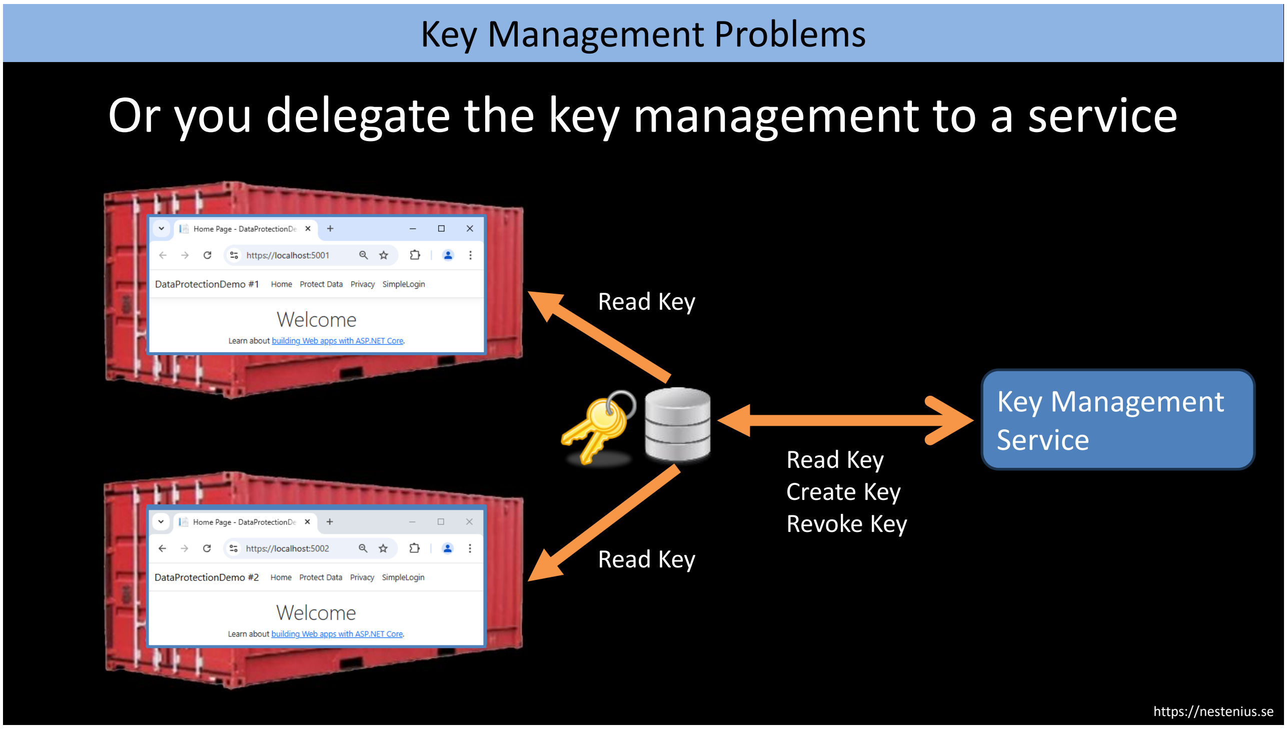
```
// Secondary
builder.Services.AddDataProtection()
   .SetApplicationName("MyApplication")
   .DisableAutomaticKeyGeneration();
```

Read Key
Create Key
Revoke Key

Read Key

https://nestenius.se

31



Key Management Problems

Or you delegate the key management to a service

Read Key

Key Management Service

Read Key
Create Key
Revoke Key

Read Key

https://nestenius.se

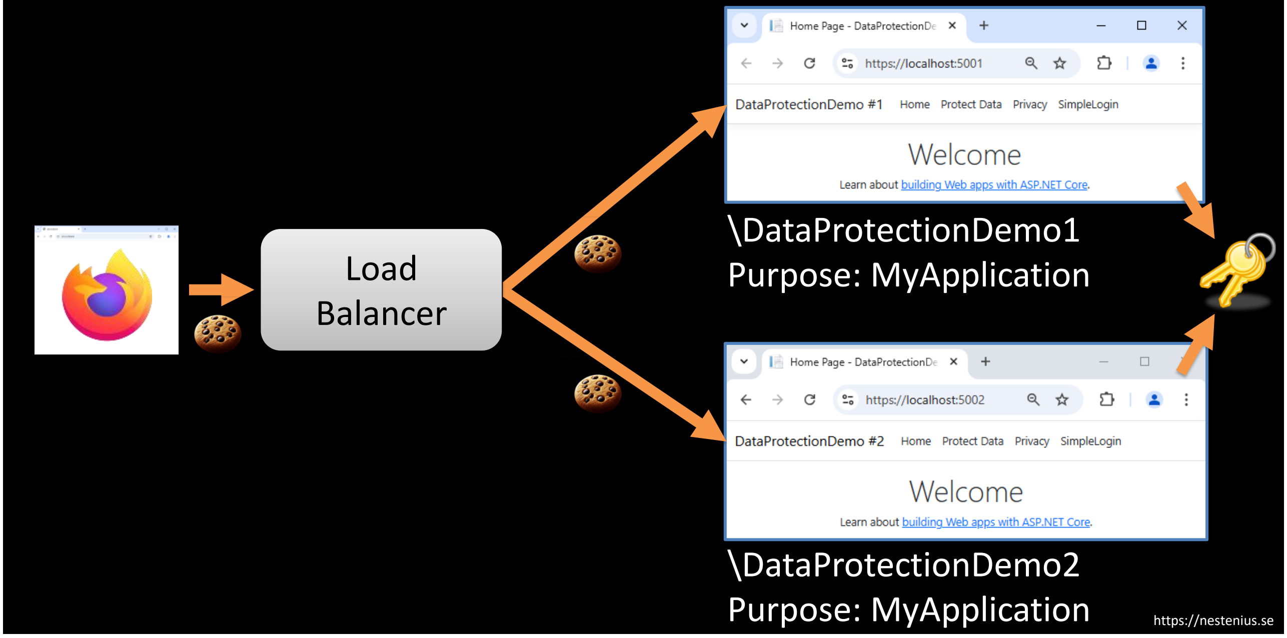
32

Summary

https://nestenius.se

33

Summary



Load Balancer

Home Page - DataProtectionDe

https://localhost:5001

DataProtectionDemo #1 Home Protect Data Privacy SimpleLogin

Welcome

Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo1
Purpose: MyApplication

Home Page - DataProtectionDe

https://localhost:5002

DataProtectionDemo #2 Home Protect Data Privacy SimpleLogin

Welcome

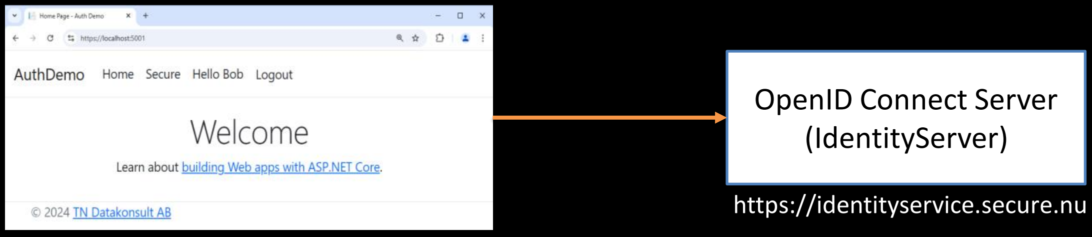
Learn about building Web apps with ASP.NET Core.

\DataProtectionDemo2
Purpose: MyApplication

https://nestenius.se

34

Part #2



Home Page - Auth Demo

https://localhost:5001

AuthDemo Home Secure Hello Bob Logout

Welcome

Learn about building Web apps with ASP.NET Core.

© 2024 TN Datakonsult AB

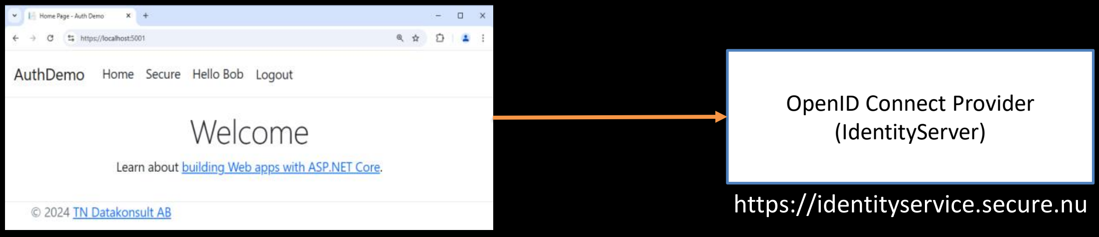
OpenID Connect Server (IdentityServer)

https://identityservice.secure.nu

https://localhost:5001

35

Problem #1 – HTTP-based Client



Home Page - Auth Demo

https://localhost:5001

AuthDemo Home Secure Hello Bob Logout

Welcome

Learn about building Web apps with ASP.NET Core.

© 2024 TN Datakonsult AB

OpenID Connect Provider (IdentityServer)

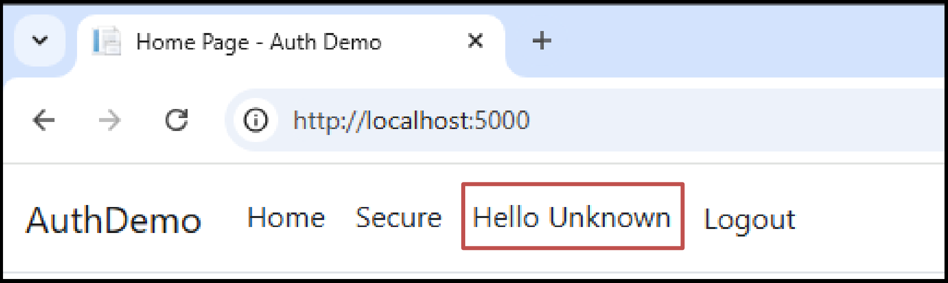
https://identityservice.secure.nu

http://localhost:5000

36

Problem #1 – HTTP-based Client

Authenticating over **HTTP** Works great!

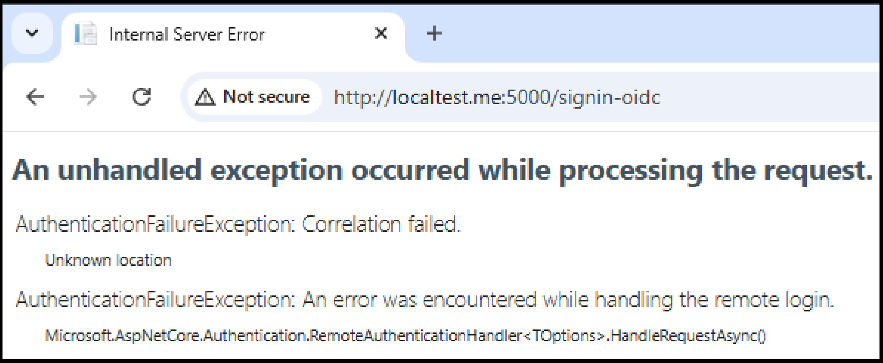


However, we have a problem!

https://nestenius.se

Problem #1 – HTTP-based Client

Authenticating over a **non-localhost** domain fails!



An unhandled exception occurred while processing the request.

AuthenticationFailureException: Correlation failed.

Unknown location

AuthenticationFailureException: An error was encountered while handling the remote login.

Microsoft.AspNetCore.Authentication.RemoteAuthenticationHandler<TOptions>.HandleRequestAsync()

Why does it work over **localhost**?

https://nestenius.se

Problem #1 – HTTP-based Client

These are considered **Potentially Trustworthy**

- `http://127.0.0.1`
- `http://localhost`
- `http://*.localhost`
- `file://`

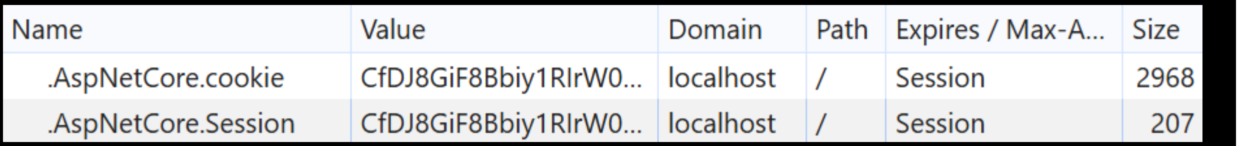
Demonstration

```
<h1>Is this context secure?</h1>
<p id="secure-status">Checking...</p>

<script>
    const isSecure = window.isSecureContext

    document.getElementById('secure-status').textContent =
        isSecure ? "Yes, this is a secure context." :
                   "No, this is not a secure context.";
</script>
```

Problem #2 – Cookie Confusion

Cookie Confusion

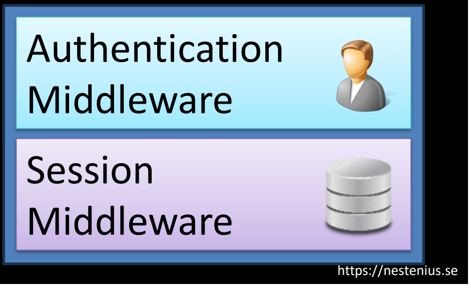
What is the difference?



Name	Value	Domain	Path	Expires / Max-A...	Size
.AspNetCore.cookie	CfDJ8GiF8Bbiy1RIrW0...	localhost	/	Session	2968
.AspNetCore.Session	CfDJ8GiF8Bbiy1RIrW0...	localhost	/	Session	207

- AspNetCore.cookie
  - User authentication session (User login)
- AspNetCore.Session
  - Temporary session data

ASP.NET Core



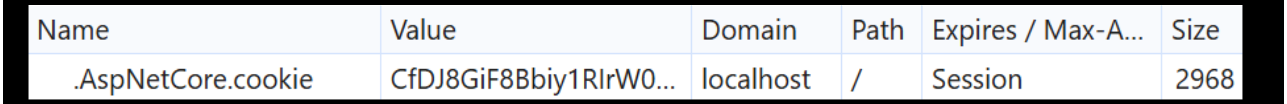
Cookies Ingredients

https://nestenius.se

43

Cookies Ingredients

What is inside the authentication cookie?

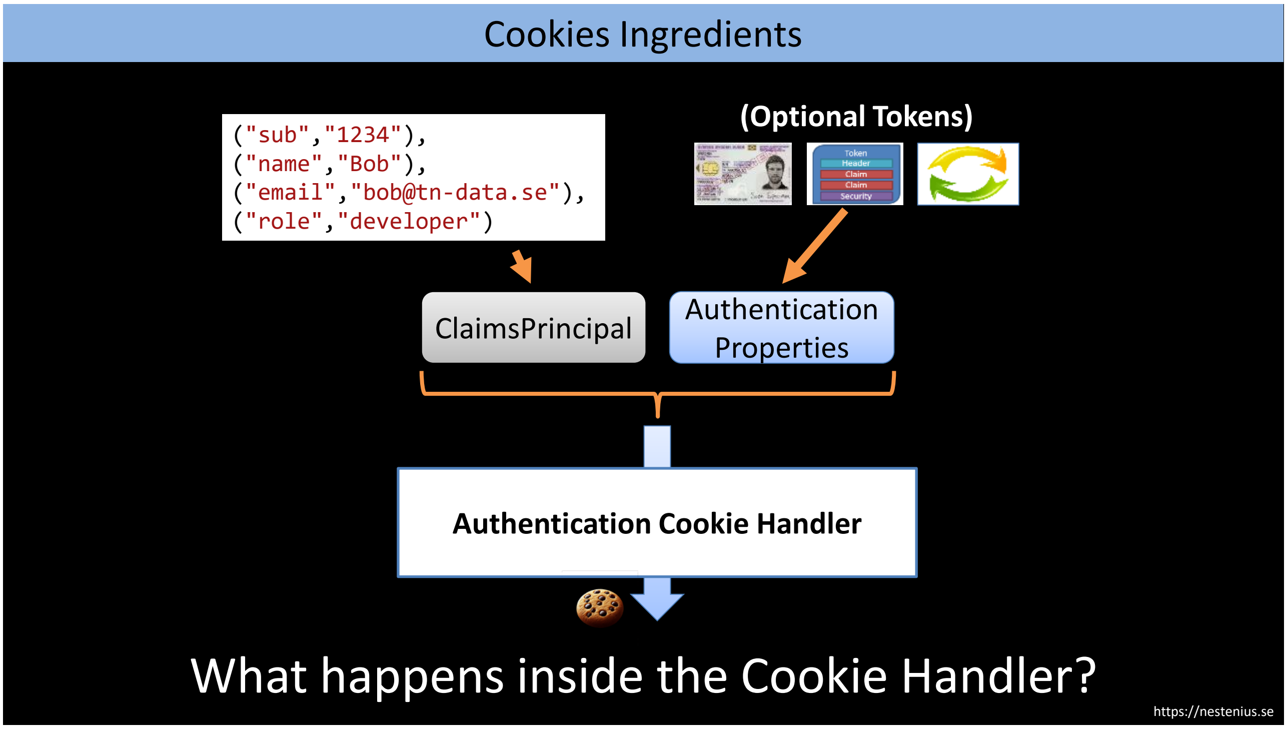


Name	Value	Domain	Path	Expires / Max-A...	Size
.AspNetCore.cookie	CfDJ8GiF8Bbiy1RlrW0...	localhost	/	Session	2968

Let's find out!

https://nestenius.se

44



Cookies Ingredients

```
("sub","1234"),
("name","Bob"),
("email","bob@tn-data.se"),
("role","developer")
```

(Optional Tokens)

ClaimsPrincipal

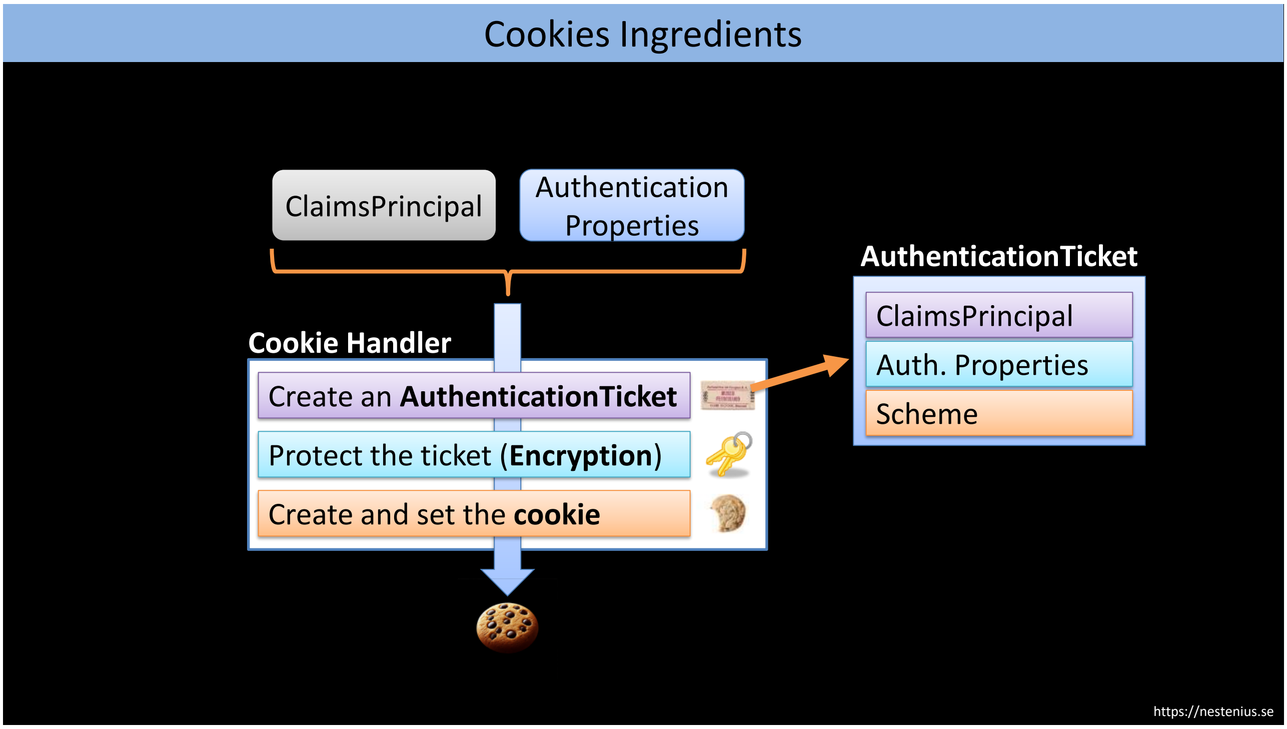
Authentication Properties

Authentication Cookie Handler

What happens inside the Cookie Handler?

https://nestenius.se

45



Cookies Ingredients

ClaimsPrincipal

Authentication Properties

Cookie Handler

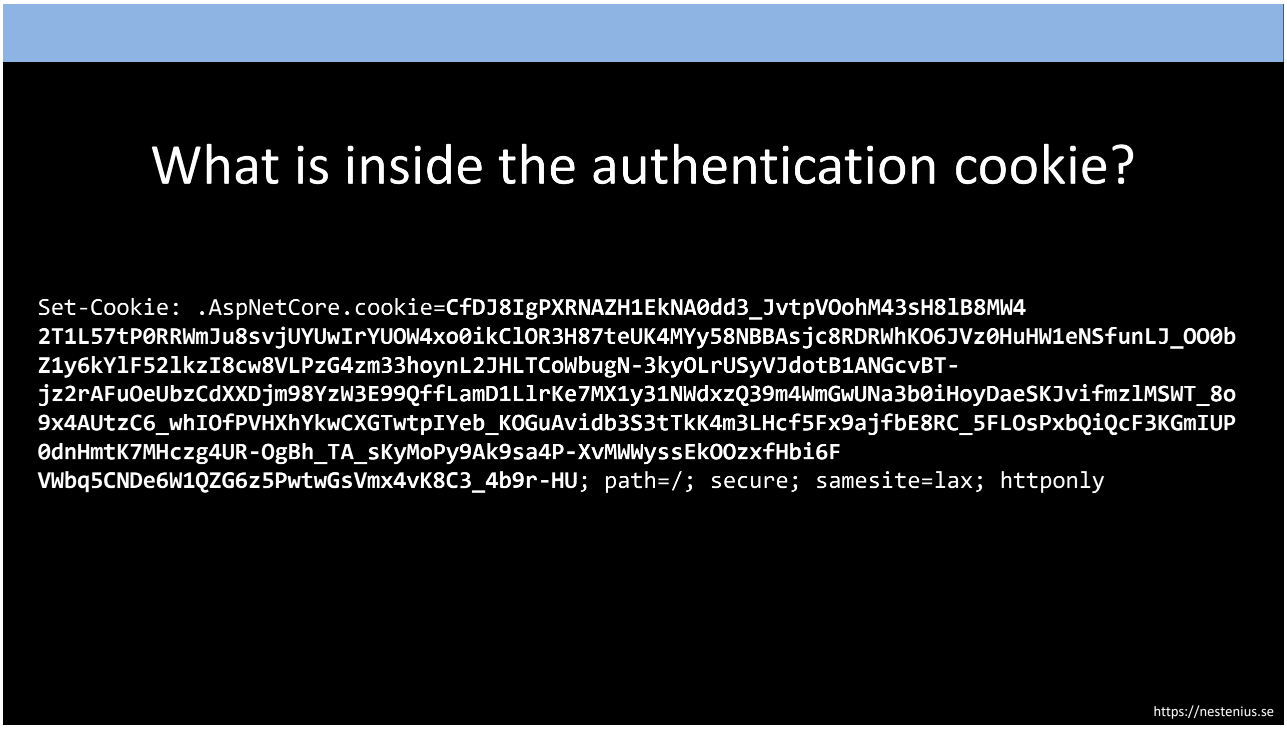
- Create an **AuthenticationTicket**
- Protect the ticket (**Encryption**)
- Create and set the **cookie**

AuthenticationTicket

- ClaimsPrincipal
- Auth. Properties
- Scheme

https://nestenius.se

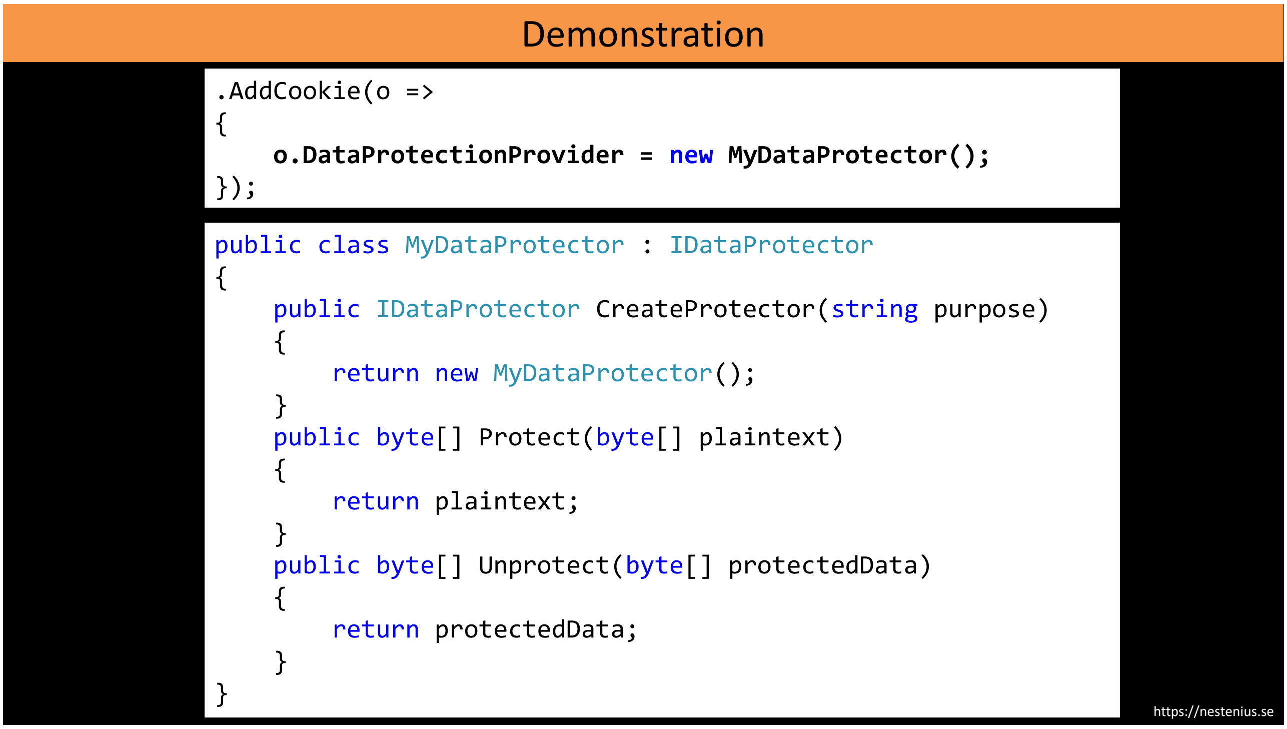
46



What is inside the authentication cookie?

```
Set-Cookie: .AspNetCore.cookie=CfDJ8IgPXRNAZH1EkNA0dd3_JvtpVOohM43sH8lB8MW4
2T1L57tP0RRWmJu8svjUYUwIrYUOW4xo0ikClOR3H87teUK4MYy58NBBAsjc8RDRWhKO6JVz0HuHW1eNSfunLJ_OO0b
Z1y6kYlF52lkzI8cw8VLPzG4zm33hoynL2JHLTCoWbugN-3kyOLrUSyVJdotB1ANGcvBT-
jz2rAFuOeUbzCdXXDjm98YzW3E99QffLamD1LlrKe7MX1y31NWdxzQ39m4WmGwUNa3b0iHoyDaeSKJvifmzlMSWT_8o
9x4AUtzC6_whIOfPVHXhYkwCXGTwtpIYeb_KOGuAvidb3S3tTkK4m3LHcf5Fx9ajfbE8RC_5FLOsPxbQiQcF3KGmIUP
0dnHmtK7MHczg4UR-OgBh_TA_sKyMoPy9Ak9sa4P-XvMWWyssEkOOzxfHbi6F
VWbq5CNDe6W1QZG6z5PwtwGsVmx4vK8C3_4b9r-HU; path=/; secure; samesite=lax; httponly
```

https://nestenius.se



Demonstration

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

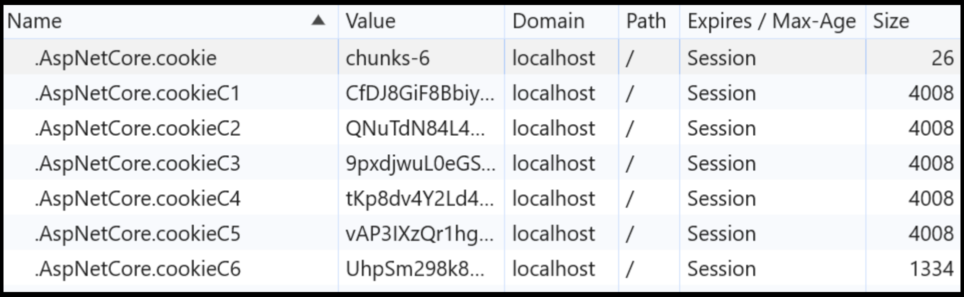
```
public class MyDataProtector : IDataProtector
{
    public IDataProtector CreateProtector(string purpose)
    {
        return new MyDataProtector();
    }
    public byte[] Protect(byte[] plaintext)
    {
        return plaintext;
    }
    public byte[] Unprotect(byte[] protectedData)
    {
  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

https://nestenius.se

Large Cookies

Large Cookies

Multiple cookies may be used to store the data

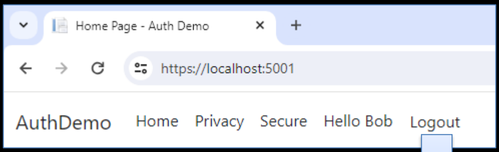


Name ▲	Value	Domain	Path	Expires / Max-Age	Size
.AspNetCore.cookie	chunks-6	localhost	/	Session	26
.AspNetCore.cookieC1	CfDJ8GiF8Bbiy...	localhost	/	Session	4008
.AspNetCore.cookieC2	QNuTdN84L4...	localhost	/	Session	4008
.AspNetCore.cookieC3	9pxdjwuL0eGS...	localhost	/	Session	4008
.AspNetCore.cookieC4	tKp8dv4Y2Ld4...	localhost	/	Session	4008
.AspNetCore.cookieC5	vAP3IXzQr1hg...	localhost	/	Session	4008
.AspNetCore.cookieC6	UhpSm298k8...	localhost	/	Session	1334

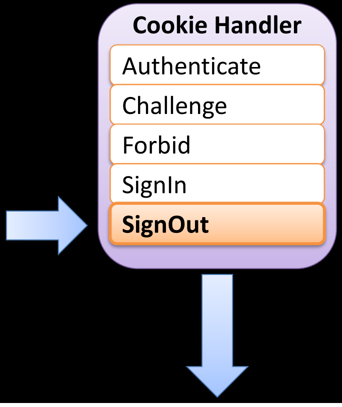
Can we improve this? Yes, we explore this later!

Insecure SignOut

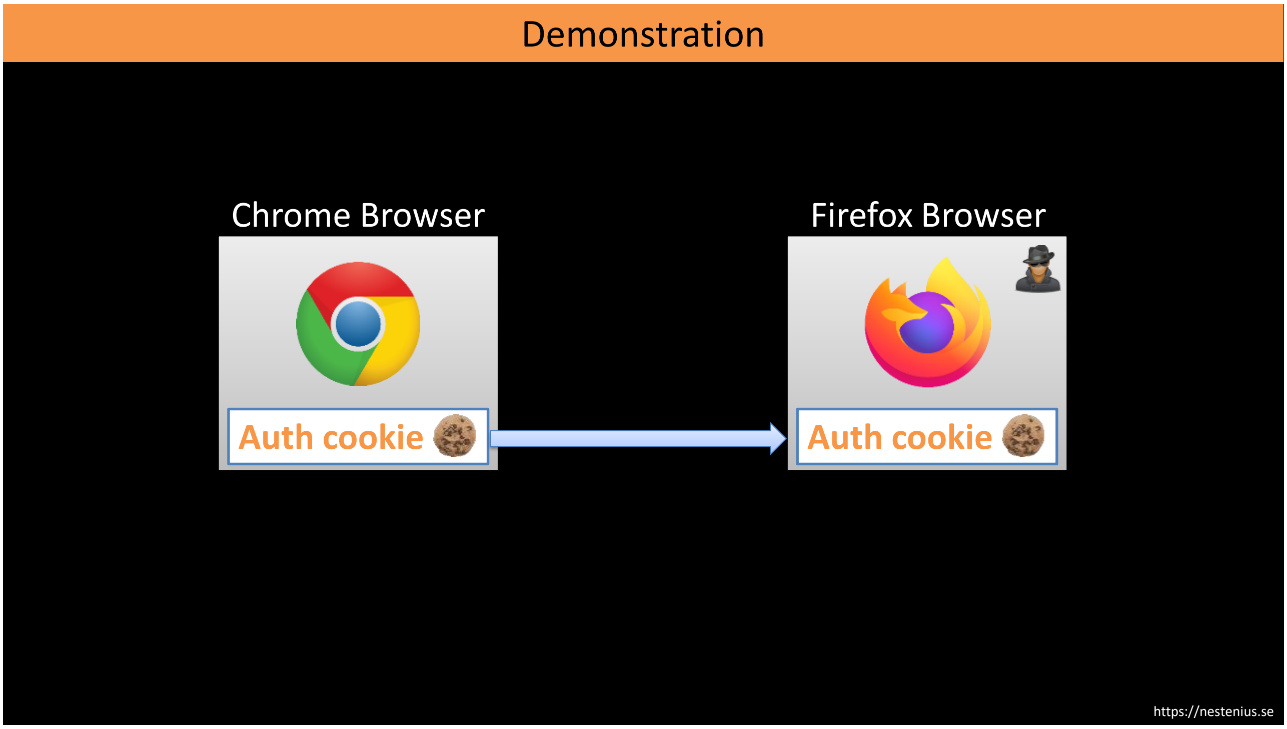
SignOut



```
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; ...
```



Demonstration

Chrome Browser

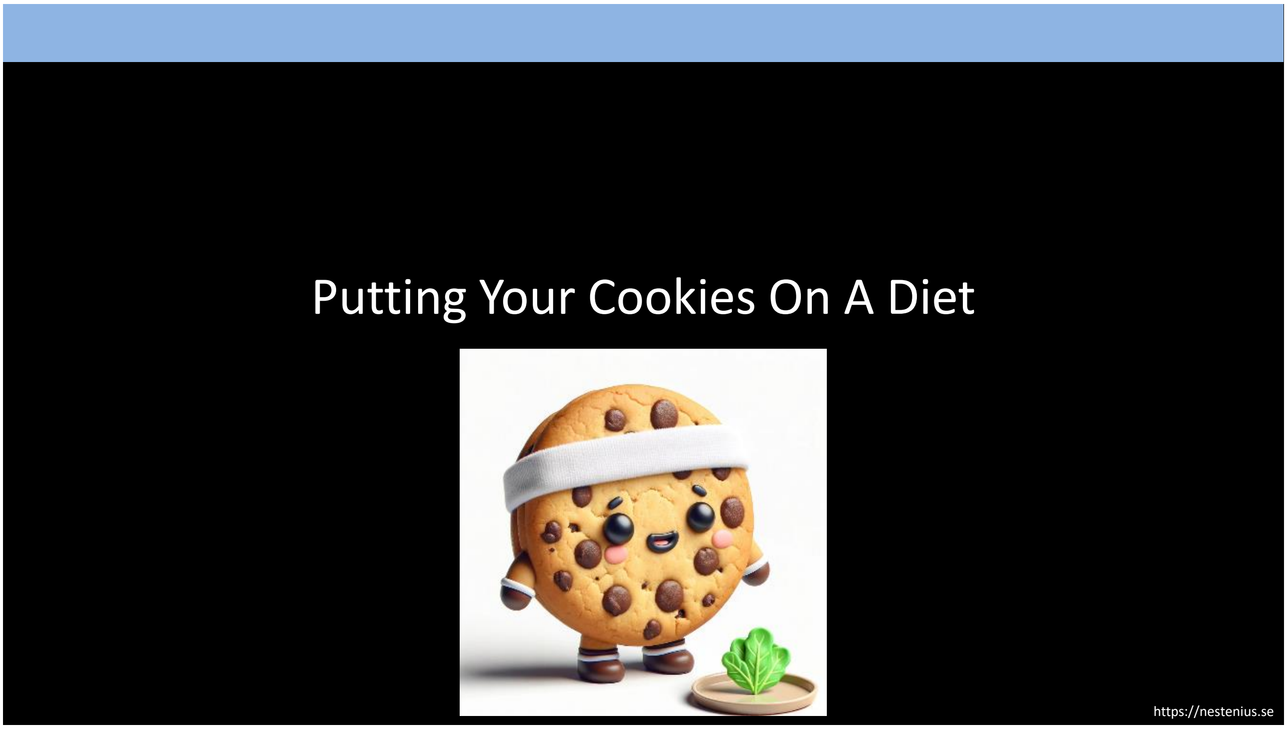
Firefox Browser

Auth cookie

Auth cookie

https://nestenius.se

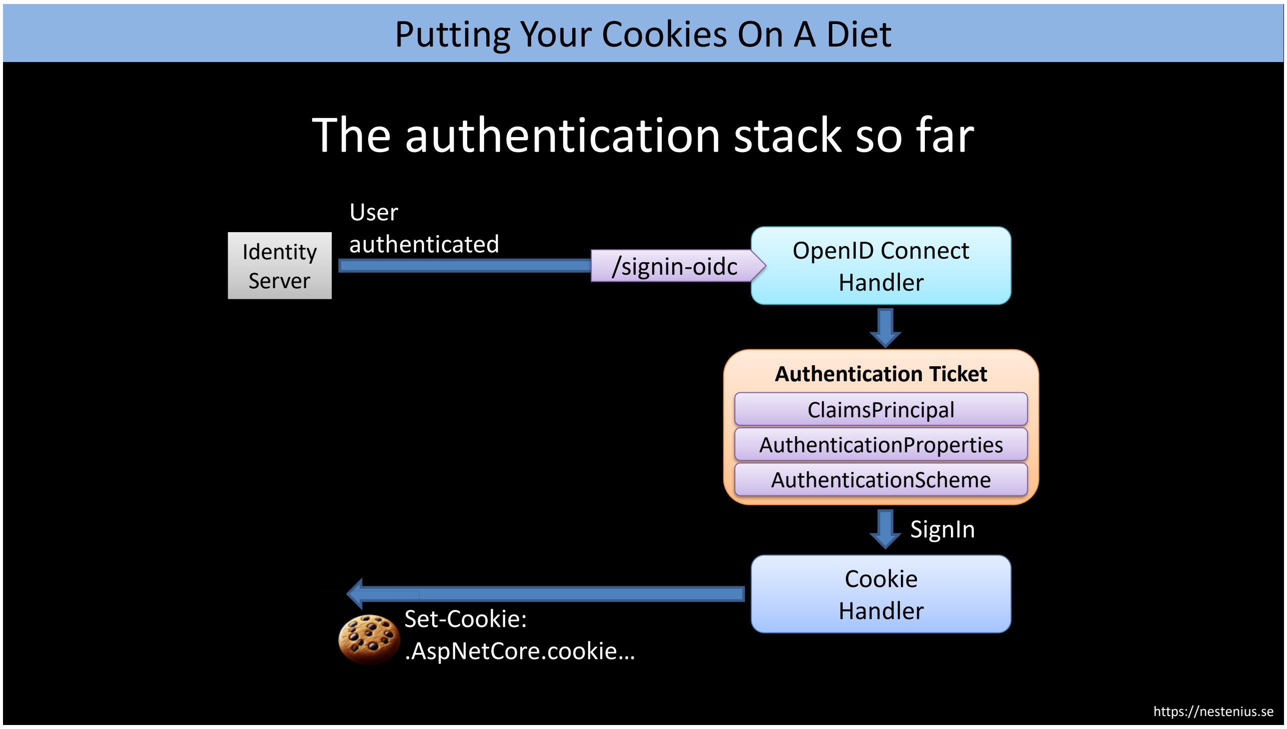
53



Putting Your Cookies On A Diet

https://nestenius.se

54



Putting Your Cookies On A Diet

The authentication stack so far

Identity Server

User authenticated

/signin-oidc

OpenID Connect Handler

Authentication Ticket

ClaimsPrincipal

AuthenticationProperties

AuthenticationScheme

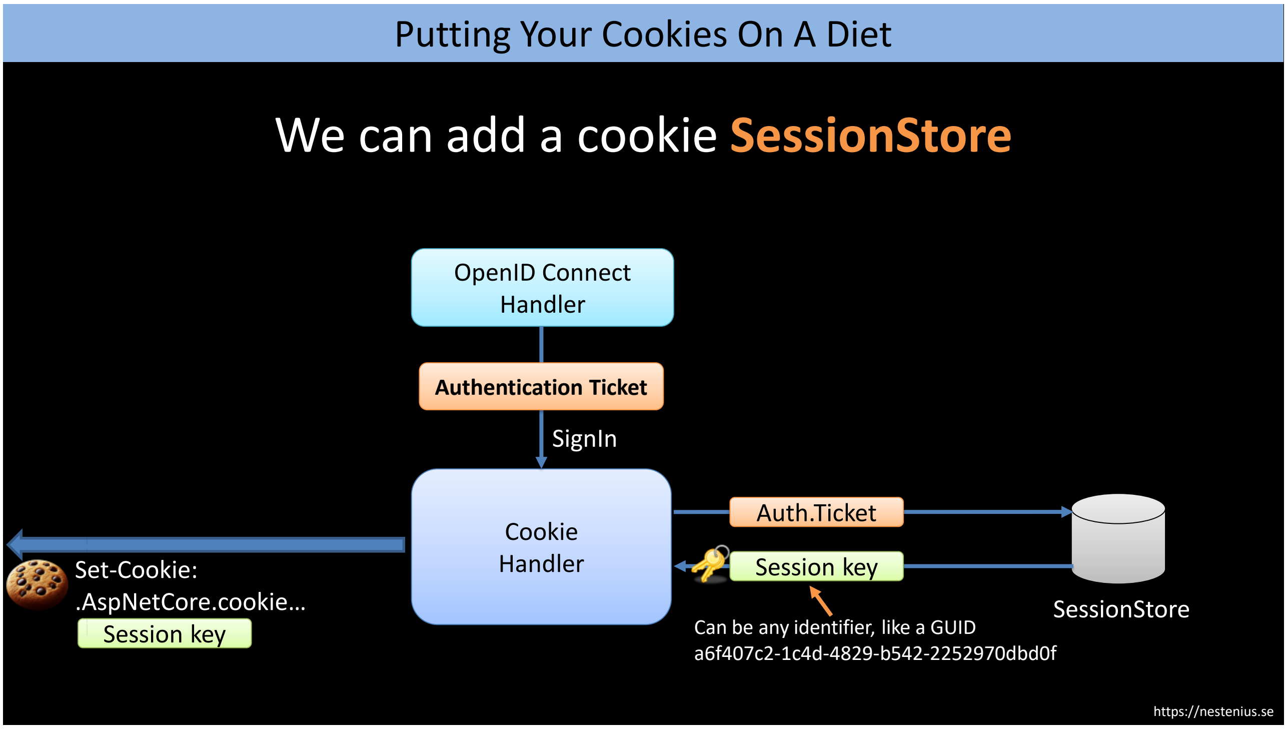
SignIn

Cookie Handler

Set-Cookie: .AspNetCore.cookie…

https://nestenius.se

55



Putting Your Cookies On A Diet

We can add a cookie **SessionStore**

OpenID Connect Handler

Authentication Ticket

SignIn

Cookie Handler

Auth.Ticket

Session key

SessionStore

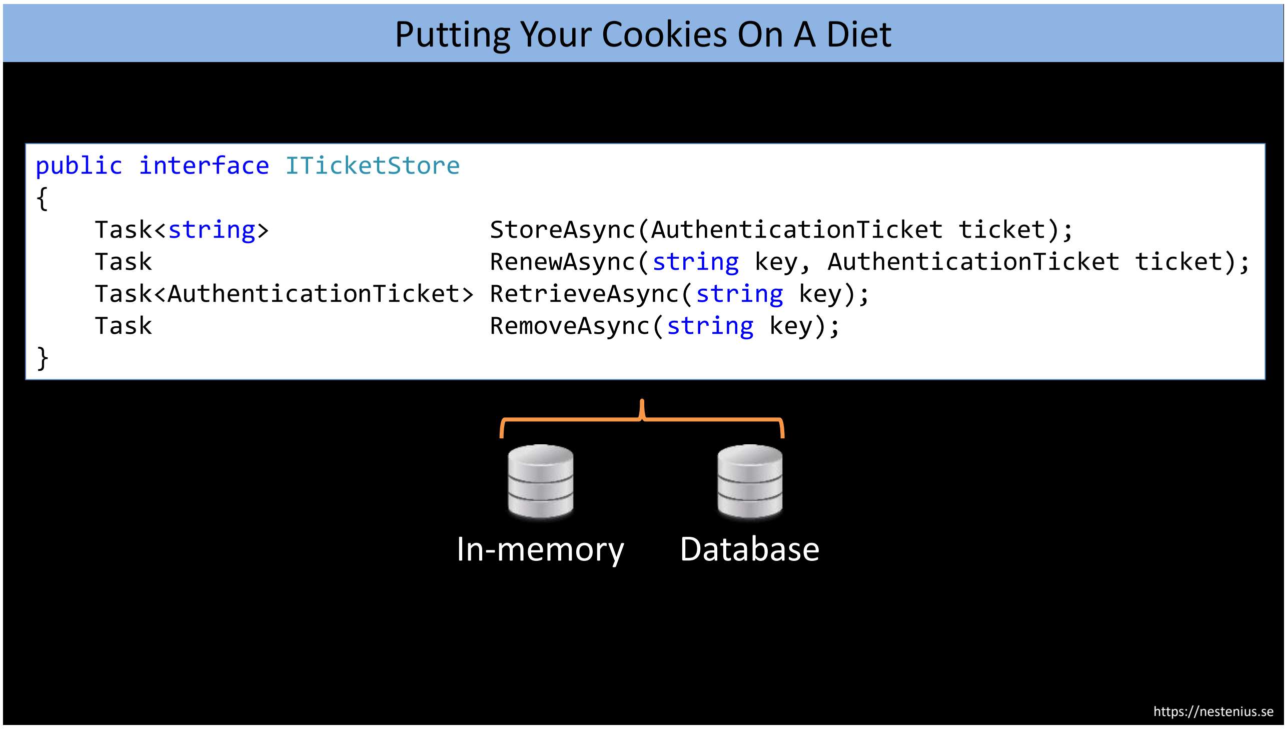
Can be any identifier, like a GUID
a6f407c2-1c4d-4829-b542-2252970dbd0f

Set-Cookie: .AspNetCore.cookie…

Session key

https://nestenius.se

56



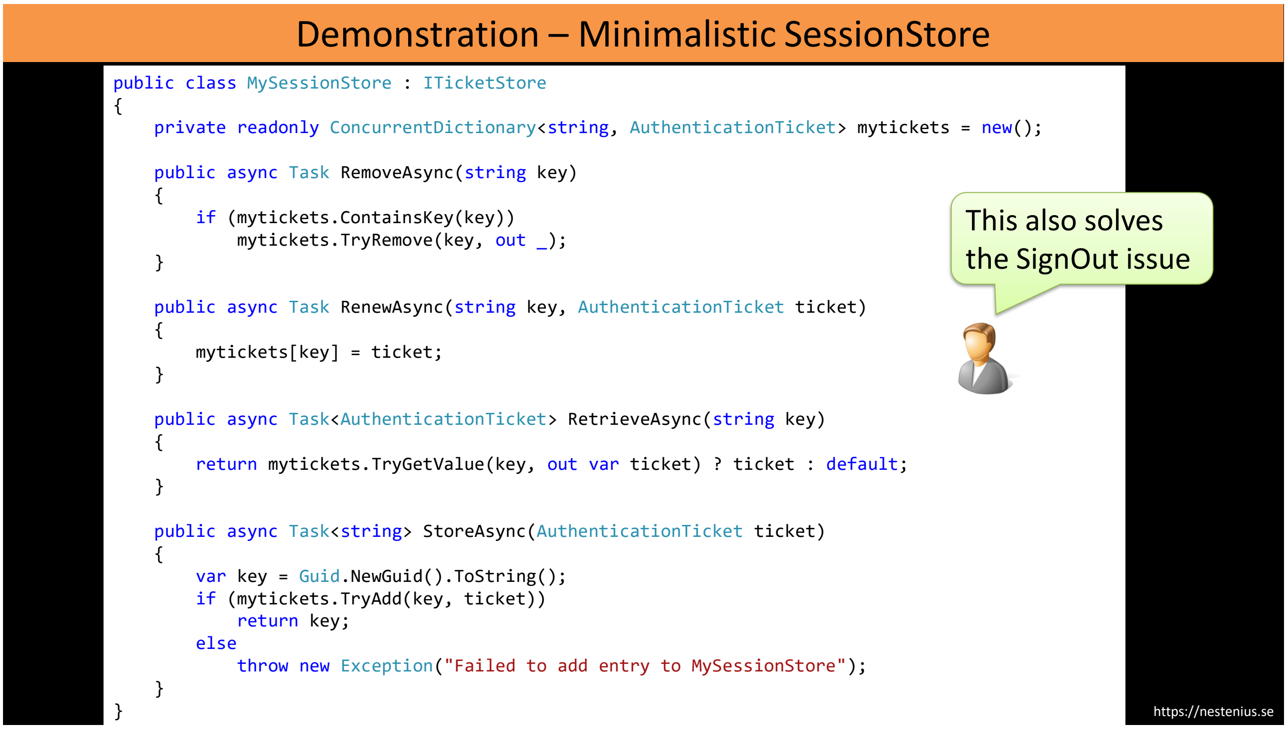
Putting Your Cookies On A Diet

```
public interface ITicketStore
{
    Task<string>                StoreAsync(AuthenticationTicket ticket);
    Task                        RenewAsync(string key, AuthenticationTicket ticket);
    Task<AuthenticationTicket> RetrieveAsync(string key);
    Task                        RemoveAsync(string key);
}
```

In-memory
Database

https://nestenius.se

57



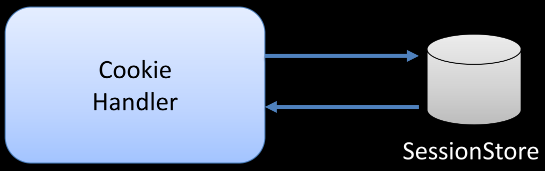
Demonstration – Minimalistic SessionStore

```
public class MySessionStore : ITicketStore
{
    private readonly ConcurrentDictionary<string, AuthenticationTicket> mytickets = new();

    public async Task RemoveAsync(string key)
    {
        if (mytickets.ContainsKey(key))
            mytickets.TryRemove(key, out _);
    }

    public async Task RenewAsync(string key, AuthenticationTicket ticket)
    {
        mytickets[key] = ticket;
    }

    public async Task<AuthenticationTicket> RetrieveAsync(string key)
    {
        return mytickets.TryGetValue(key, out var ticket) ? ticket : default;
    }

    public async Task<string> StoreAsync(AuthenticationTicket ticket)
    {
        var key = Guid.NewGuid().ToString();
        if (mytickets.TryAdd(key, ticket))
            return key;
        else
            throw new Exception("Failed to add entry to MySessionStore");
    }
}
```

https://nestenius.se

58

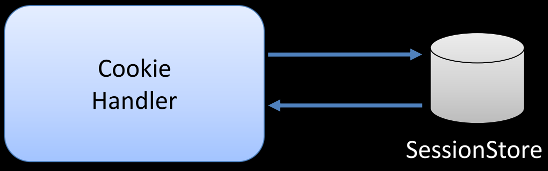
Possibilities using a Session Store

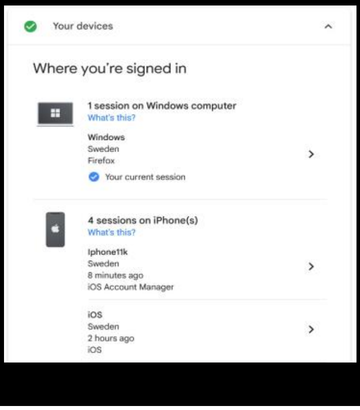


https://nestenius.se

59

Possibilities using a Session Store





Your devices

Where you're signed in

1 session on Windows computer
What's this?

Windows
Sweden
Firefox
Your current session

4 sessions on iPhone(s)
What's this?

Iphone11k
Sweden
8 minutes ago
iOS Account Manager

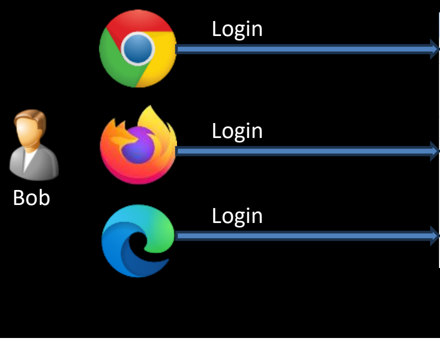
iOS
Sweden
2 hours ago
iOS

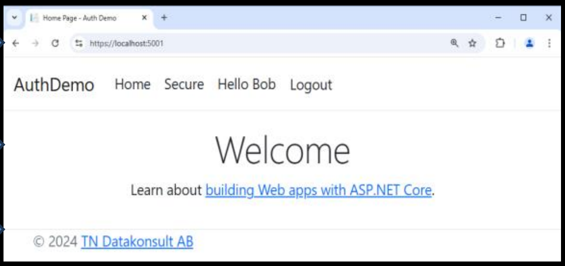
https://nestenius.se

60

Demonstration – Advanced SessionStore

```
.AddCookie("cookie", o =>
{
   ...
   o.SessionStore = new AdvancedSessionStore();

})
```





AuthDemo Home Secure Hello Bob Logout

Welcome

Learn about building Web apps with ASP.NET Core.

© 2024 TN Datakonsult AB

Insecure Redirects

Insecure Redirect

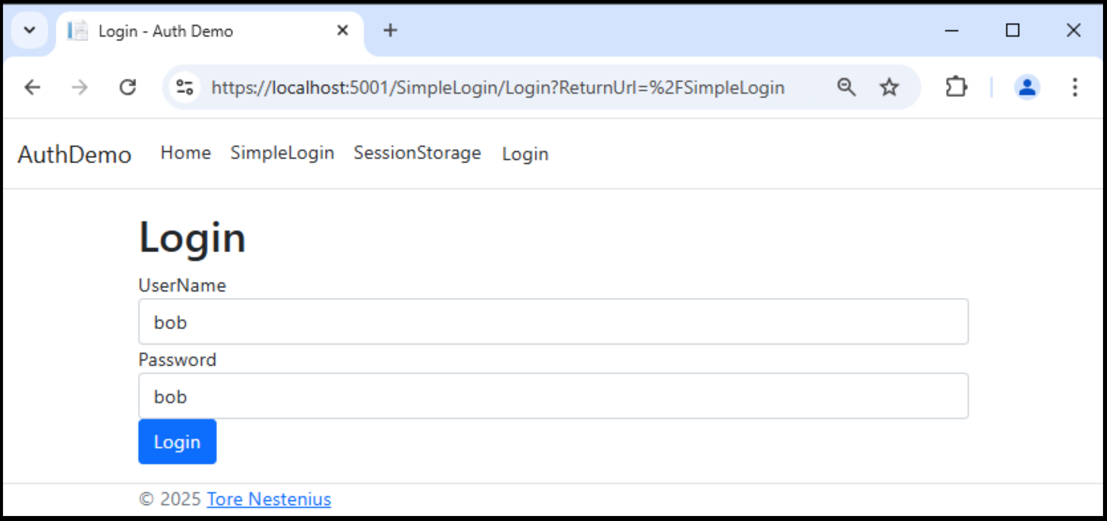
What is the problem?

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginModel loginCredentials)
{
    var claims = new List<Claim>()
        { new("sub","1234"), new("name","Bob")};

    var identity = new ClaimsIdentity(claims, "pwd");
    var principal = new ClaimsPrincipal(identity);

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

    return Redirect(loginCredentials.ReturnUrl);
}
```

https://nestenius.se

63

Demonstration – Introducing LocalRedirect


Login - Auth Demo
https://localhost:5001/SimpleLogin/Login?ReturnUrl=%2FSimpleLogin
AuthDemo
Home
SimpleLogin
SessionStorage
Login

Login

UserName

bob

Password

bob

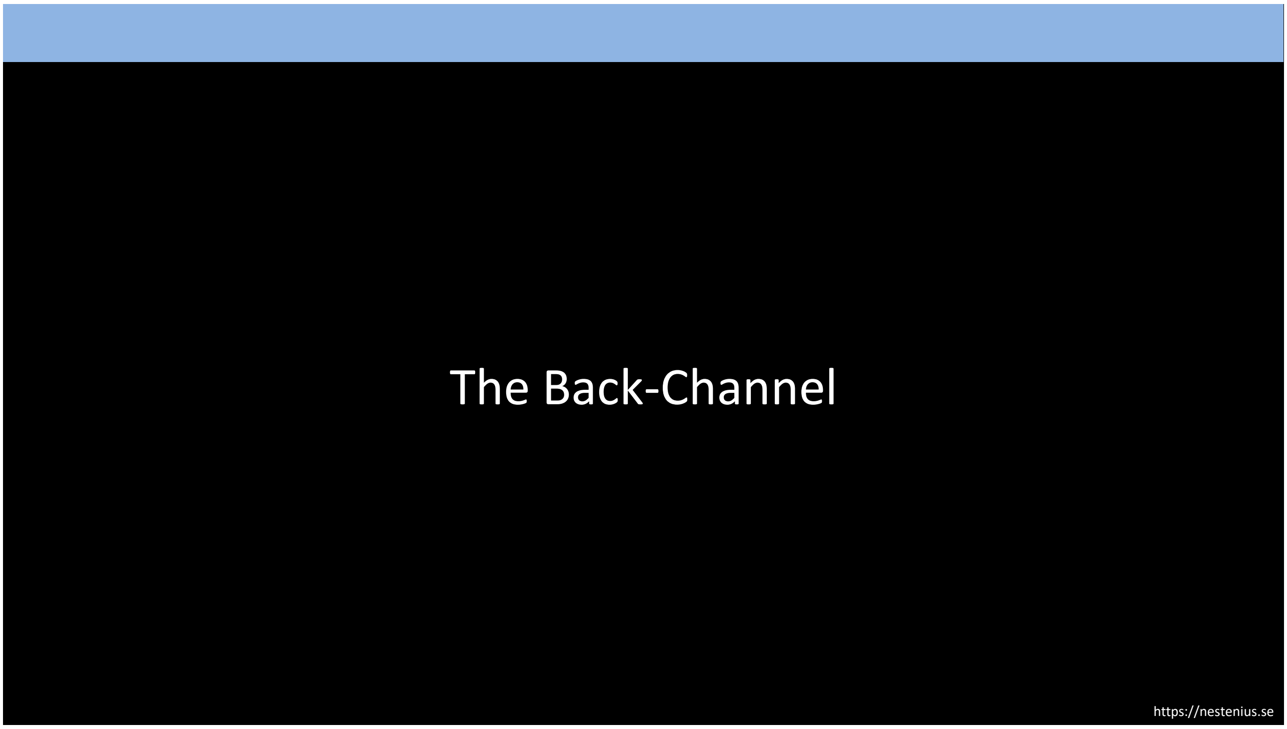
Login

© 2025 Tore Nestenius

```
sêʧụsŋ ĿộçắľŖêđîsêçʧ ľộĝîŋCsêđêŋʧîắľş ŖêʧụsŋÛsľ
```

https://nestenius.se

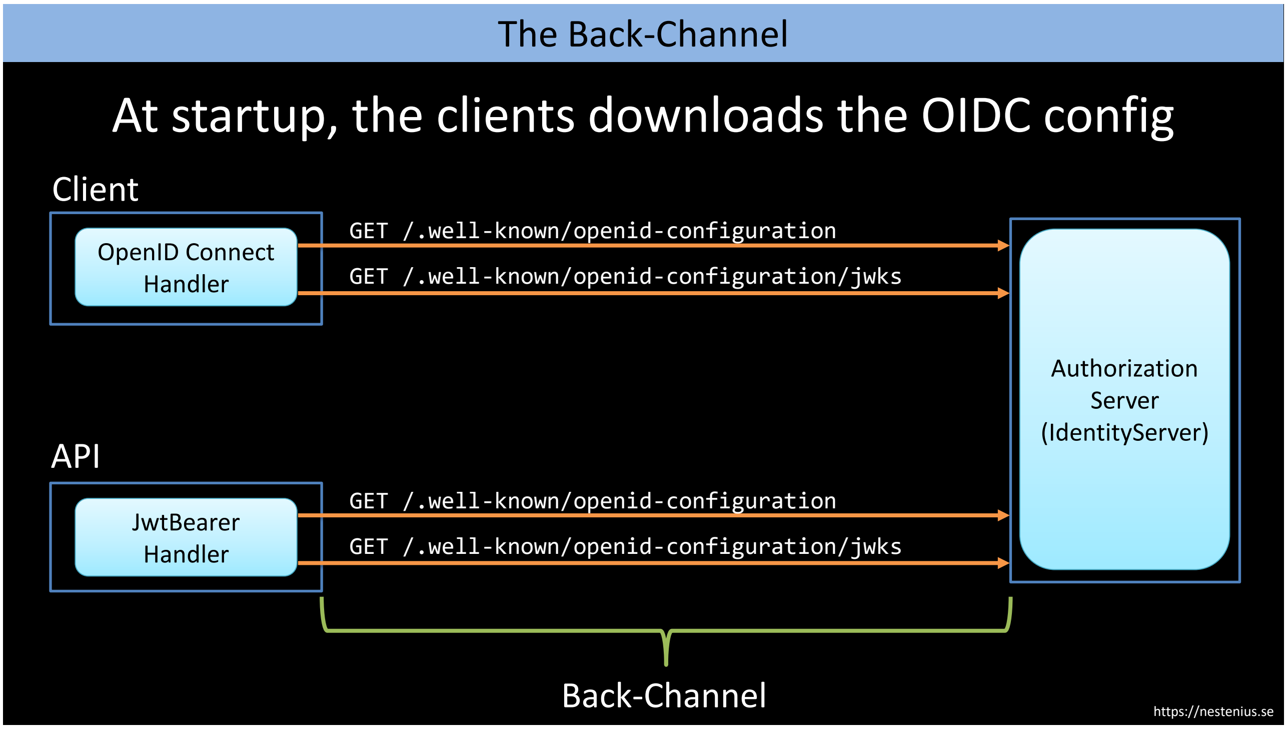
64



The Back-Channel

https://nestenius.se

65



The Back-Channel

At startup, the clients downloads the OIDC config

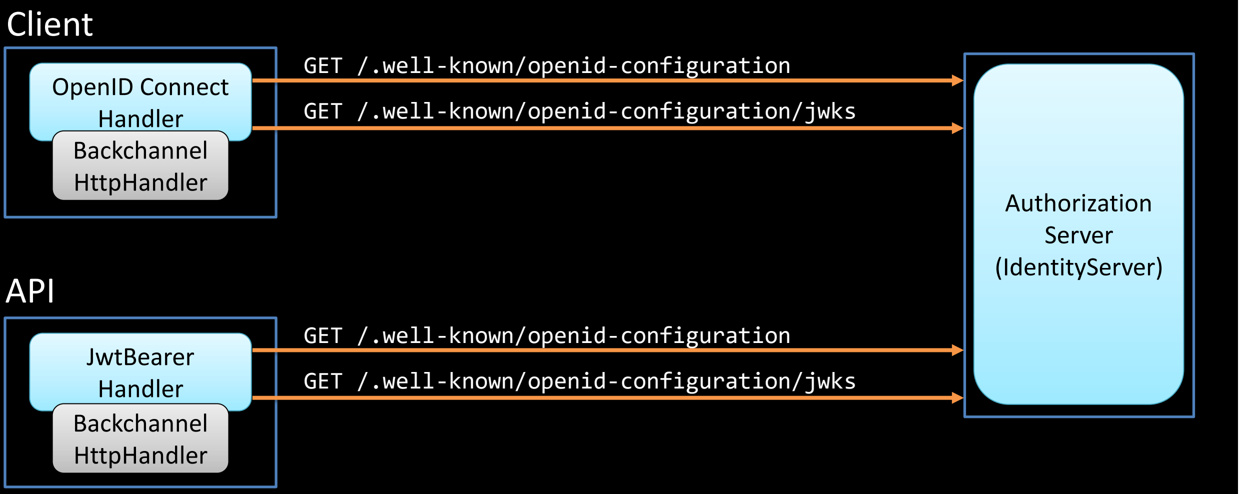
Client
OpenID Connect Handler
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration/jwks
API
JwtBearer Handler
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration/jwks
Authorization Server (IdentityServer)
Back-Channel

https://nestenius.se

66

The Back-Channel

This traffic can be captured


Client
OpenID Connect Handler
Backchannel HttpHandler
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration/jwks
API
JwtBearer Handler
Backchannel HttpHandler
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration/jwks
Authorization Server (IdentityServer)

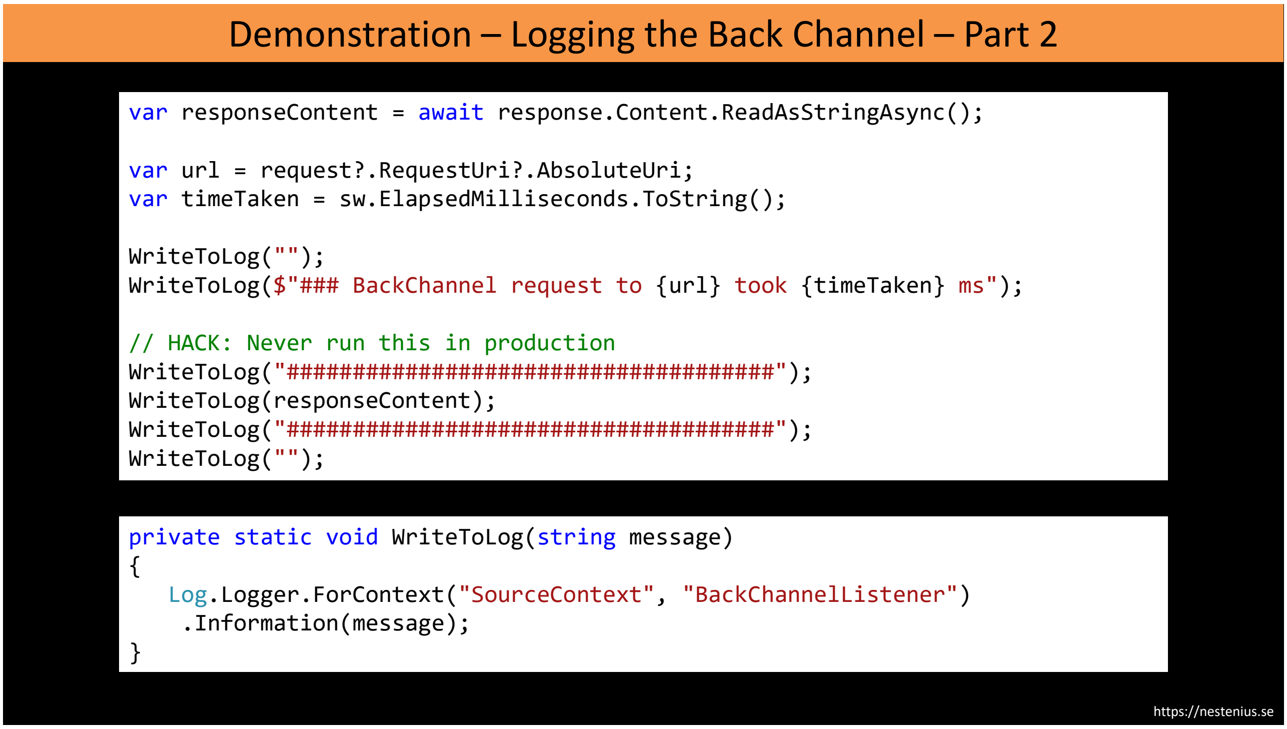
https://nestenius.se

Demonstration – Logging the Back-Channel

```
public class BackChannelListener : DelegatingHandler
{
    public BackChannelListener() : base(new HttpClientHandler())
    { }

    protected async override Task<HttpResponseMessage> SendAsync(HttpRequestMessage request,
                                                              CancellationToken token)
    {
        var sw = new Stopwatch();
        sw.Start();

        var response = await base.SendAsync(request, token);

        sw.Stop();

        // TODO...

        return response;
    }
}
```

```
ộ BắçlçḥắŋŋêľĦŧŧřĤắŋđľês   ŋêx BắçlCḥắŋŋêľĽîşŧĝêŋês
```

https://nestenius.se

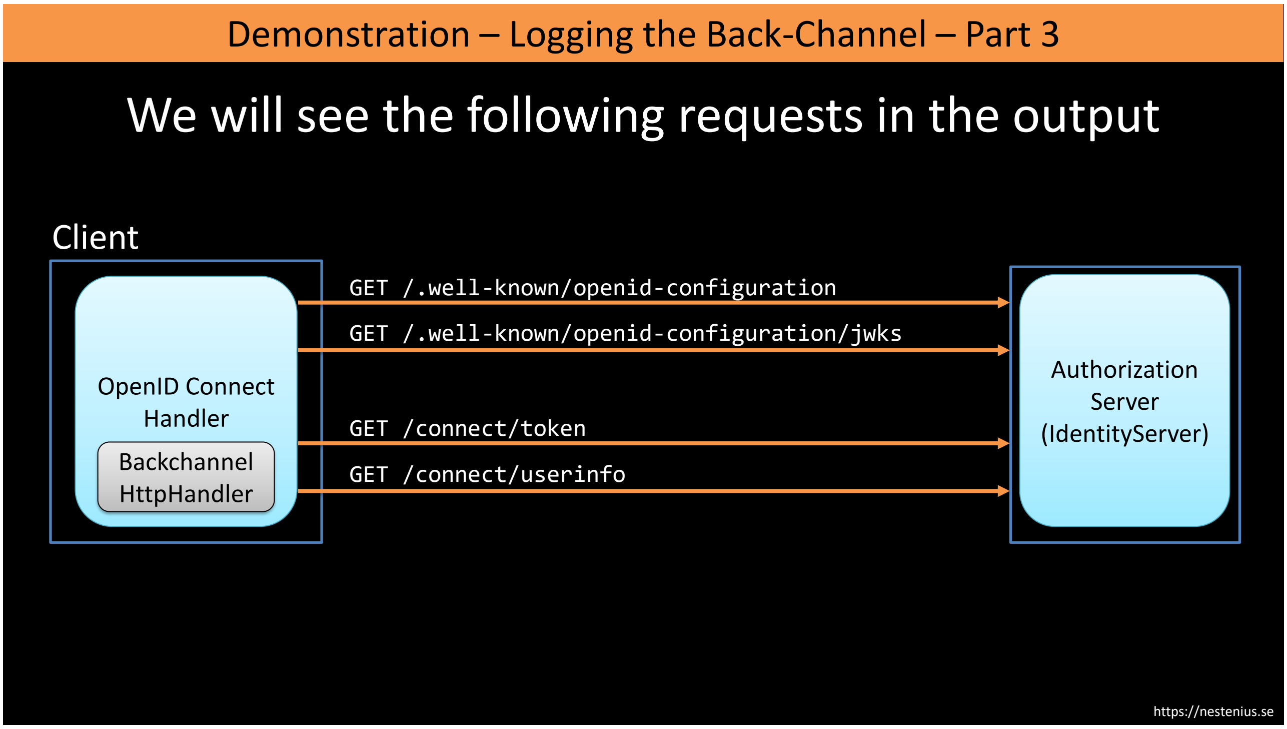


Demonstration – Logging the Back Channel – Part 2

```
var responseContent = await response.Content.ReadAsStringAsync();

var url = request?.RequestUri?.AbsoluteUri;
var timeTaken = sw.ElapsedMilliseconds.ToString();

WriteToLog("");
WriteToLog($"### BackChannel request to {url} took {timeTaken} ms");

// HACK: Never run this in production
WriteToLog("#####################################");
WriteToLog(responseContent);
WriteToLog("#####################################");
WriteToLog("");
```

```
private static void WriteToLog(string message)
{
   Log.Logger.ForContext("SourceContext", "BackChannelListener")
    .Information(message);
}
```

https://nestenius.se



Demonstration – Logging the Back-Channel – Part 3

We will see the following requests in the output

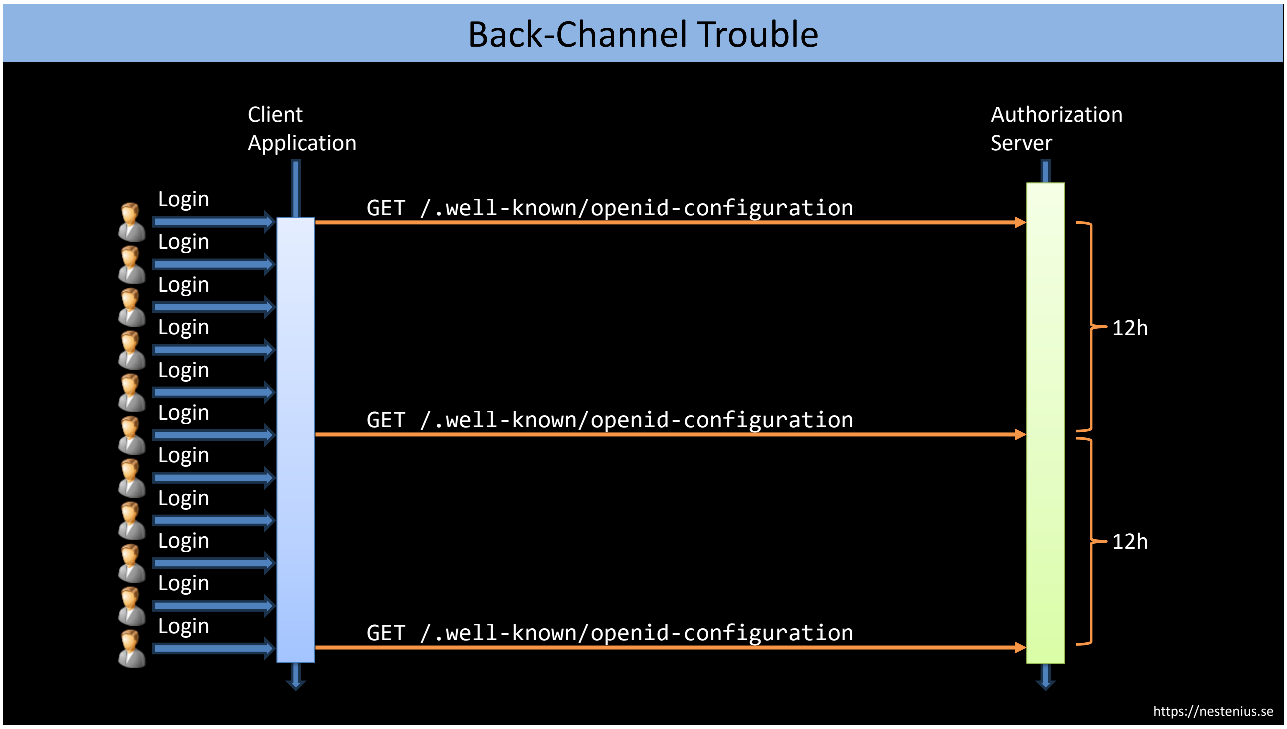
Client
OpenID Connect Handler
Backchannel HttpHandler
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration/jwks
GET /connect/token
GET /connect/userinfo
Authorization Server (IdentityServer)

https://nestenius.se

Back-Channel Trouble

https://nestenius.se

71



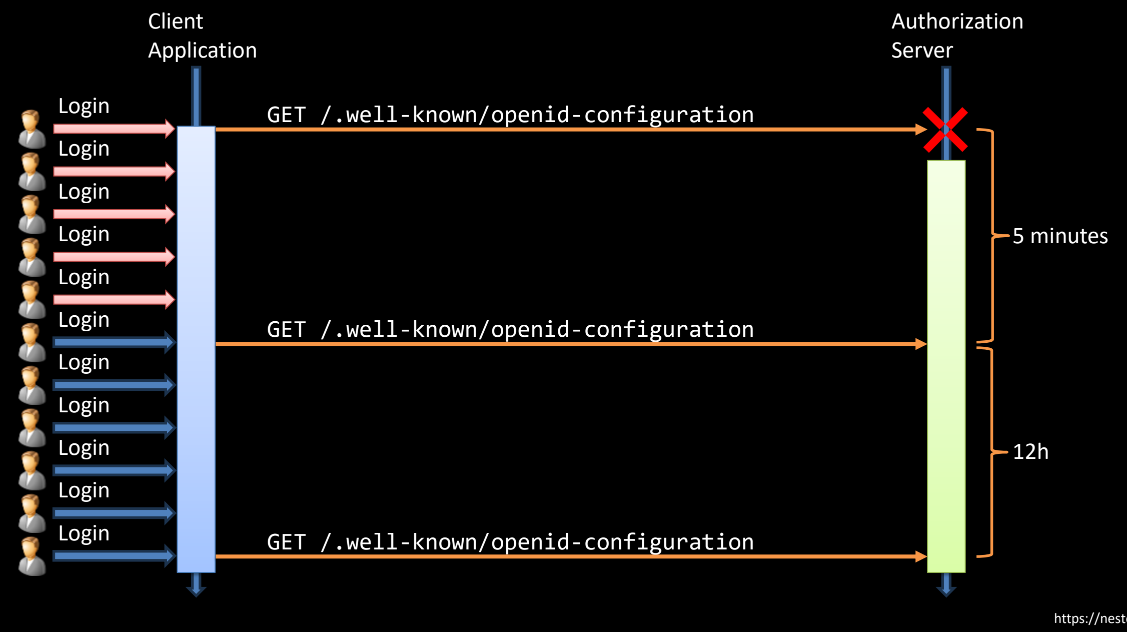
Back-Channel Trouble

Client Application
Authorization Server
Login
Login
Login
Login
Login
Login
Login
Login
Login
Login
Login
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration
12h
12h

https://nestenius.se

72

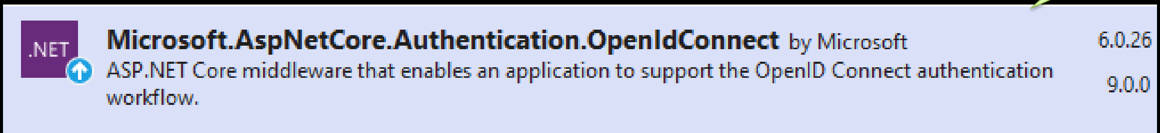
Back-Channel Trouble


Client
Application
Authorization
Server
Login
Login
Login
Login
Login
Login
Login
Login
Login
Login
Login
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration
GET /.well-known/openid-configuration
5 minutes
12h

https://nestenius.se

What changed in Version 6.0.26?

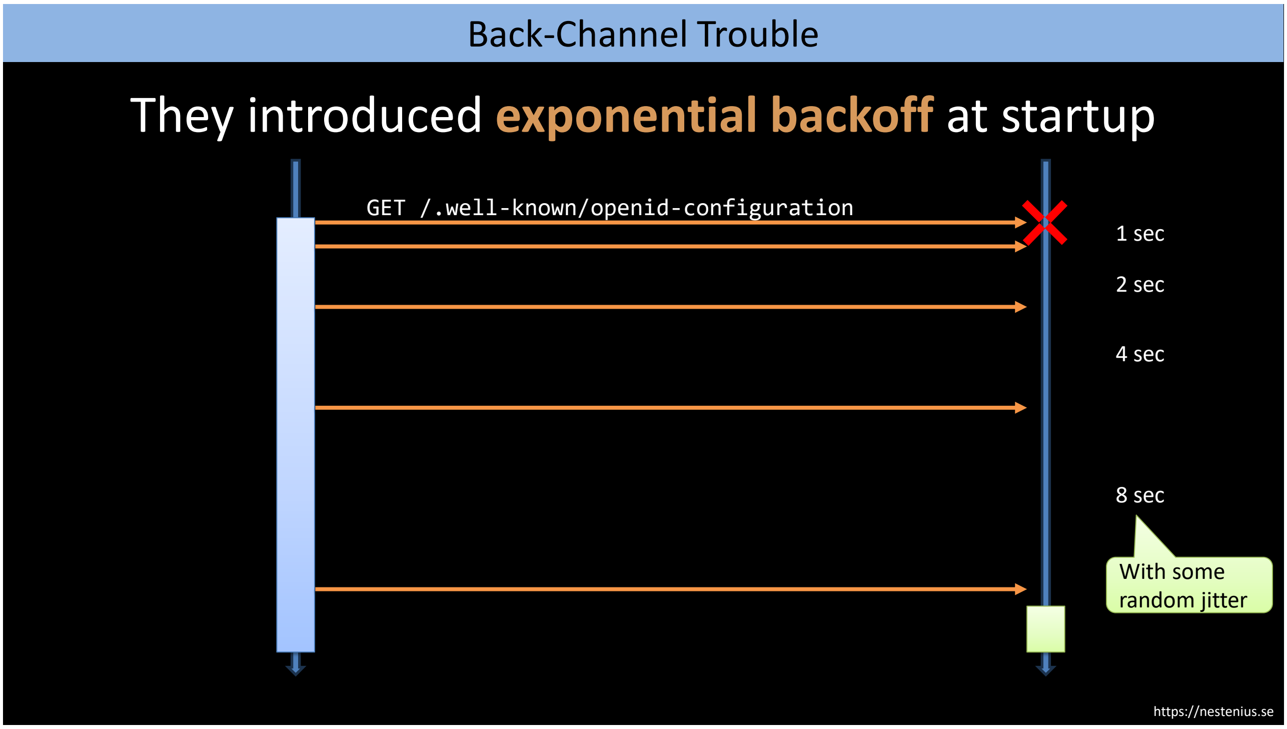
September 2024



Microsoft.AspNetCore.Authentication.OpenIdConnect by Microsoft
ASP.NET Core middleware that enables an application to support the OpenID Connect authentication workflow.

6.0.26
9.0.0

https://nestenius.se



Back-Channel Trouble

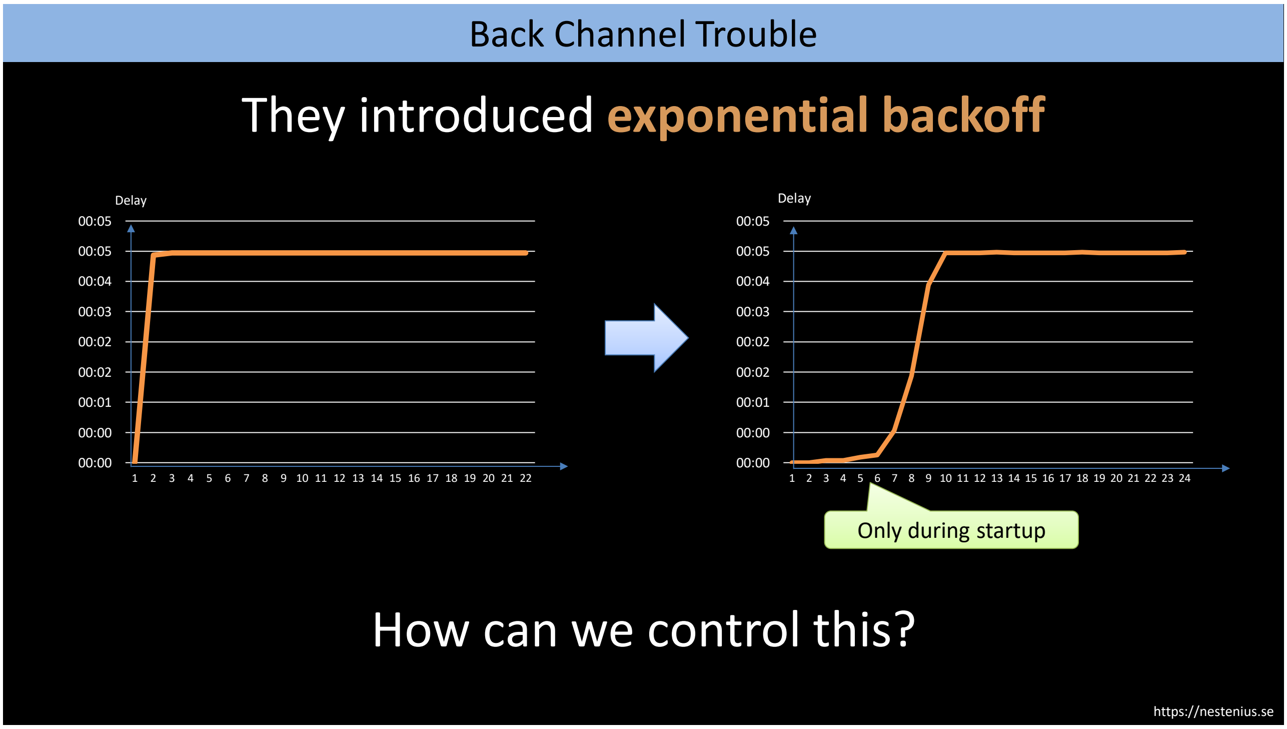
They introduced **exponential backoff** at startup

`GET /.well-known/openid-configuration`

With some random jitter

https://nestenius.se

75



Back Channel Trouble

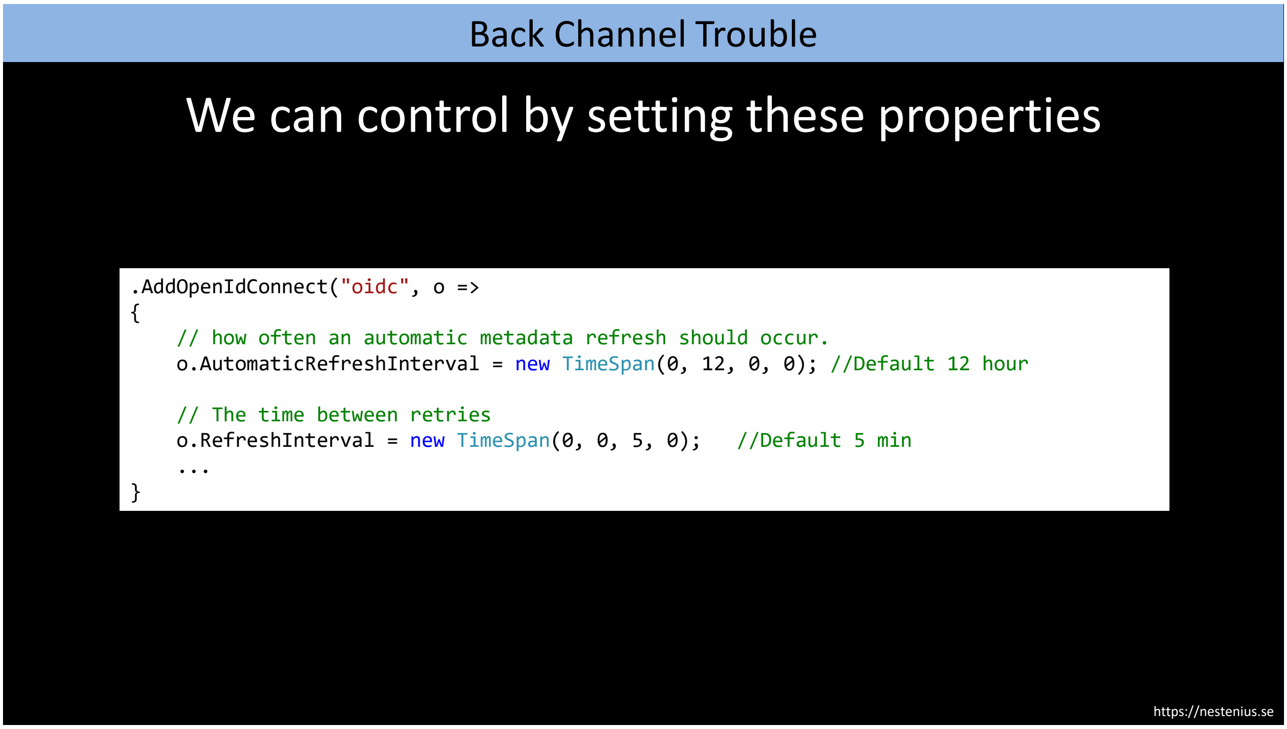
They introduced **exponential backoff**

Only during startup

How can we control this?

https://nestenius.se

76

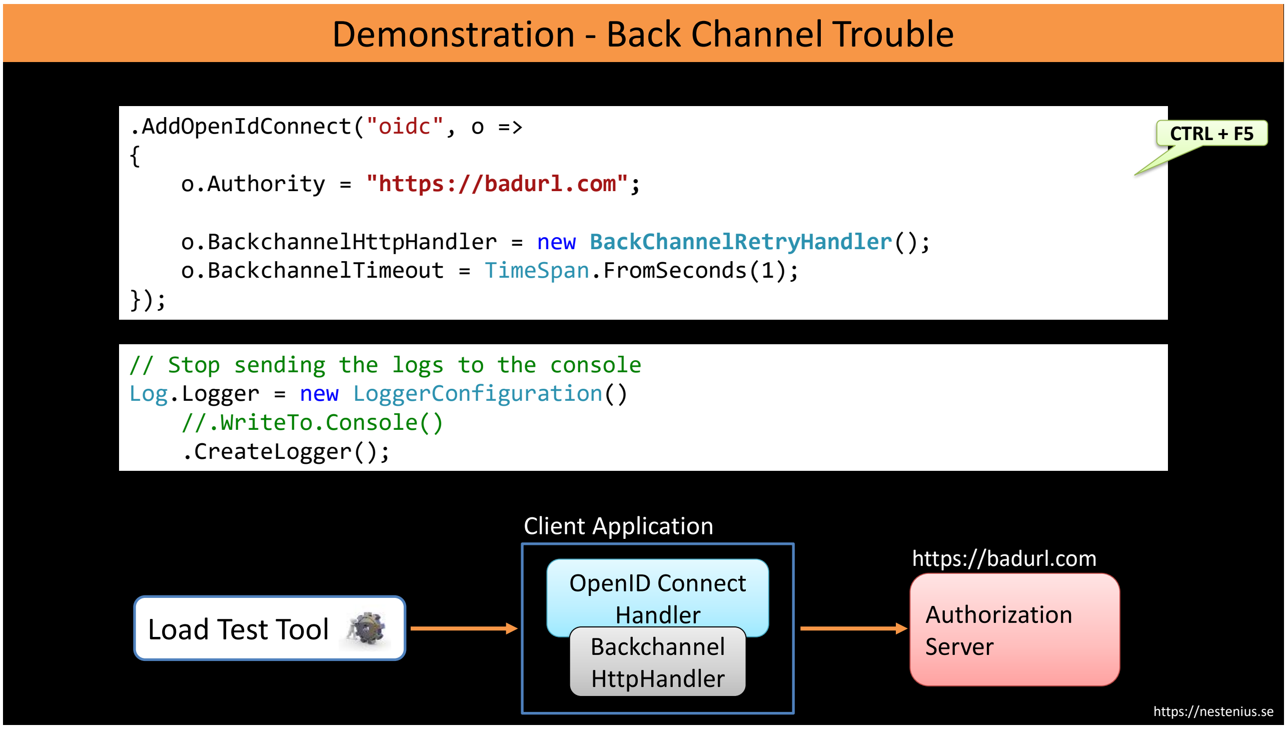


Back Channel Trouble

We can control by setting these properties

```
.AddOpenIdConnect("oidc", o =>
{
    // how often an automatic metadata refresh should occur.
    o.AutomaticRefreshInterval = new TimeSpan(0, 12, 0, 0); //Default 12 hour

    // The time between retries
    o.RefreshInterval = new TimeSpan(0, 0, 5, 0);   //Default 5 min
    ...
}
```

https://nestenius.se



Demonstration - Back Channel Trouble

```
.AddOpenIdConnect("oidc", o =>
{
    o.Authority = "https://badurl.com";

    o.BackchannelHttpHandler = new BackChannelRetryHandler();
    o.BackchannelTimeout = TimeSpan.FromSeconds(1);
});
```

CTRL + F5

```
// Stop sending the logs to the console
Log.Logger = new LoggerConfiguration()
    //.WriteTo.Console()
    .CreateLogger();
```

https://nestenius.se

QUESTIONS?

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

Blog
https://nestenius.se

Work
https://tn-data.se

https://nestenius.se