Introduction to the Data Protection API

ASP.NET Identity

Authorization

Authentication

Data Protection

https://www.tn-data.se

1

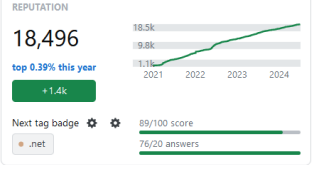
About Your Instructor

- Name: Tore Nestenius
- Programming for over 40 years
- **1996, www.programmersheaven.com**
  A popular website for programmers with over 750,000 monthly visitors.

- **2010, Cofounded of Edument AB**
  A consulting and training company.

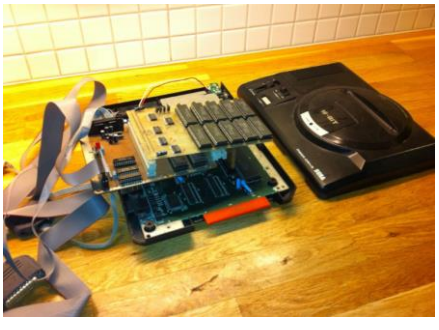
Edument

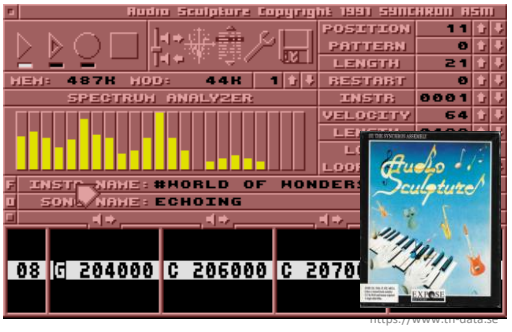
- **2020, Stack Overflow**
  Started to help others!



2

Past Projects – 1987-1993

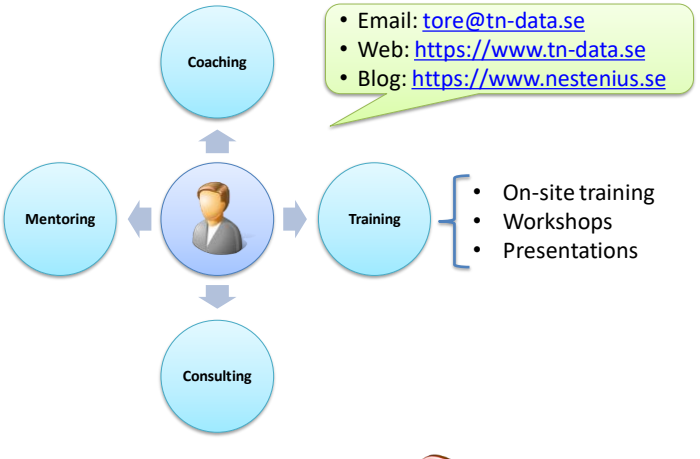




https://www.tn-data.se

My current occupation

Today, I am self-employed at T.N. Datakonsult AB.



What topics do I focus on?

https://www.tn-data.se

My training 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**
- **IdentityServer** in Production

Web
- **Web Security** Fundamentals

.NET
- Applied **D**omain-**D**riven **D**esign in .NET
- **CQRS** and **Event Sourcing** in .NET
- Building ASP.**NET Core** APIs
- **Asynchronous** Programming in C#
- **C# Expert**
- C# Fundamentals
- TDD.NET

For course details
https://tn-data.se

My blog

I blog at https://nestenius.se/

- Default Azure Credentials Under the Hood
- Improving ASP.NET Core Security By Putting Your Cookies On A Diet
- Demystifying OpenID Connect's State and Nonce Parameters
- Exploring what is inside the ASP.NET Core cookies
- Debugging cookie problems in ASP.NET Core
- BearerToken: The new Authentication handler in .NET 8
- Debugging JwtBearer Claim Problems in ASP.NET Core
- Debugging OpenID Connect Claim Problems in ASP.NET Core
- Troubleshooting JwtBearer authentication problems in ASP.NET Core
- IdentityServer – IdentityResource vs. ApiResource vs. ApiScope
- ASP.NET Core JwtBearer library: what's new?
- How I built my own Sega Mega Drive hardware dev kit from scratch
- .NET 5 Source Generators – MediatR – CQRS – OMG!
- Storing the ASP.NET Core Data Protection Key Ring in Azure Key Vault
- Exploring the non-nullable type warnings in C# 8

https://www.tn-data.se

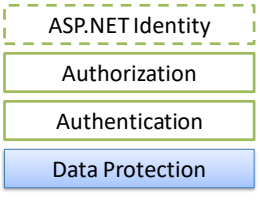
Important Notes!

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

https://www.tn-data.se

7

Introduction to the Data Protection API

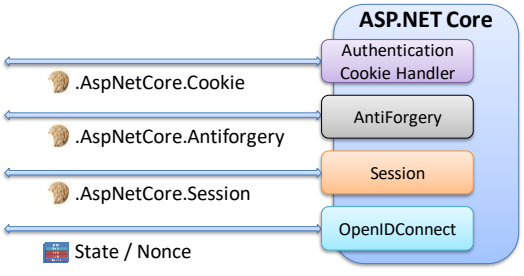


https://www.tn-data.se

8

The Data Protection API

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

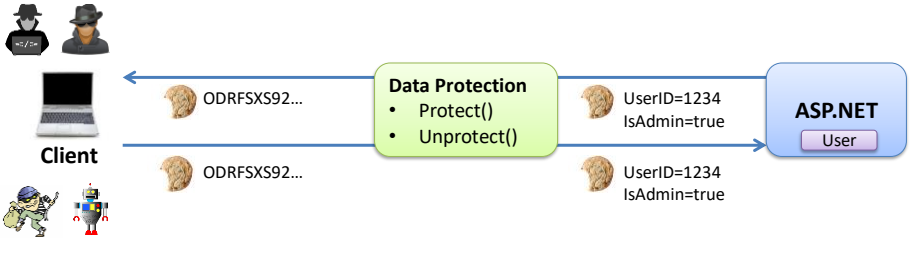


How are they secured?

https://www.tn-data.se

The Data Protection API

The cookies are secured using **encryption**



This protects them from:

- Tampering
- Eavesdropping

https://www.tn-data.se

Services provided

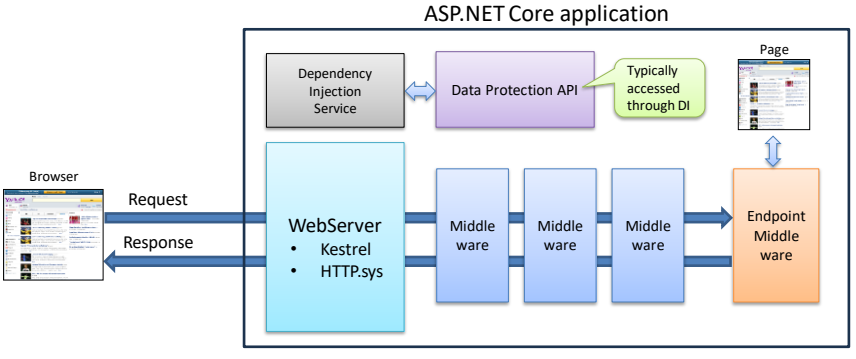
Services provided

The **Data Protection API** provides the following **services**:

- **Protect data**
  - Encryption (Protect)
  - Decryption (Unprotect)
- **Key management**
  - Key rotation
  - Key revocation

The authentication sub-system

DPAPI is implemented as a set of **services**



It is used by many features in ASP.NET Core

How do we add it to our application?

The authentication sub-system

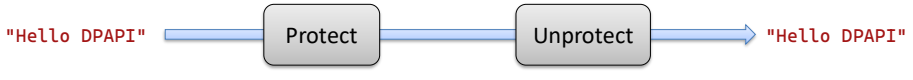
It is automatically added, when we call one of these:

```
builder.Services.AddAntiforgery();
builder.Services.AddAuthentication();
builder.Services.AddSession();
builder.Services.AddControllersWithViews();
builder.Services.AddMvc();
...
```

We can also explicitly add it using:

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

Protecting data



Protecting data

DPAPI is automatically added in most ASP.NET projects

```
public class MyController : Controller
{
    private readonly IDataProtectionProvider dataProtection;

    public ProtectDataController(IDataProtectionProvider dataProtection)
    {
        this.dataProtection = dataProtection;
    }

    public IActionResult Index()
    {
        ...
    }
}
```

We just need to ask the **DI system** for an instance of it

How do we use it?

Protecting data

The first step is to create a **protector**

```
public class MyController : Controller
{
    private readonly IDataProtectionProvider dataProtection;

    public ProtectDataController(IDataProtectionProvider dataProtection)
    {
        this.dataProtection = dataProtection;
    }

    public IActionResult Index()
    {
        var protector = dataProtection.CreateProtector(purpose: "MyPurpose");
        ...
    }
```

Why do we need to provide a **purpose**?

https://www.tn-data.se

17

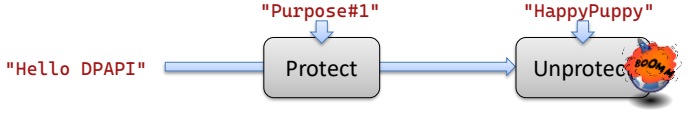
Protecting data

The **purpose** acts like a "namespace"

```
var protector = dataProtection.CreateProtector(purpose: "MyPurpose");
```

It needs to be identical for it to function correctly

Data can't be unprotected without the correct purpose



https://www.tn-data.se

18

Protecting data

The next step is to encrypt the data by calling **Protect**

```
public class MyController : Controller
{
    private readonly IDataProtectionProvider dataProtection;

    public ProtectDataController(IDataProtectionProvider dataProtection)
    {
        this.dataProtection = dataProtection;
    }

    public IActionResult Index()
    {
        var protector = dataProtection.CreateProtector(purpose: "MyPurpose");

        string encryptedData = protector.Protect("Hello DPAPI");
```

```
CfDJ8NJ06rCkv-5OufulOD3dWOr0c20Ay7AZGfEI0_zKn6hQU-
HPILXWRYBM0URCXfSlltHW0PX0OJdhiIY6vfFq_AsLiMjukiTNPLQva-
NBlLAVgf3HoShnA4pqfpF9u7ESixQHQLLqP4W_opmYRWFrq5Y
```

```
        …
    }
}
```

How do we decrypt the data?

https://www.tn-data.se

Protecting data

To decrypt the data:

```
public class MyController : Controller
{
    private readonly IDataProtectionProvider dataProtection;

    public ProtectDataController(IDataProtectionProvider dataProtection)
    {
        this.dataProtection = dataProtection;
    }

    public IActionResult Index(CryptoModel model)
    {
        var protector = dataProtection.CreateProtector("MyPurpose");

        string encryptedData = protector.Protect("Hello DPAPI");
```

We only work with strings

```
CfDJ8NJ06rCkv-5OufulOD3dWOr0c20Ay7AZGfEI0_zKn6hQU-
HPILXWRYBM0URCXfSlltHW0PX0OJdhiIY6vfFq_AsLiMjukiTNPLQva-
NBlLAVgf3HoShnA4pqfpF9u7ESixQHQLLqP4W_opmYRWFrq5Y
```

```
        string decryptedData = protector.Unprotect(encryptedData);
```

```
Hello DPAPI
```

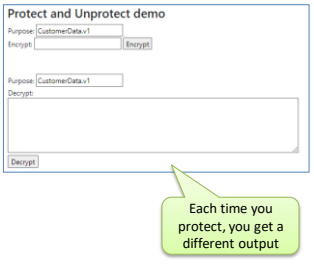
```
        …
    }
```

https://www.tn-data.se

DEMO TIME!

Demonstrating data protection

```
[HttpPost]
public IActionResult Encrypt(CryptoModel model)
{
    var _protector = dataProtection.CreateProtector(model.EncryptPurpose);

    model.EncryptedData = _protector.Protect(model.DataToEncrypt);
    model.DataToDecrypt = model.EncryptedData;

    return RedirectToAction("Index", model);
}


[HttpPost]
public IActionResult Decrypt(CryptoModel model)
{
    model.Exception = "";

    try
    {
        var _protector = dataProtection.CreateProtector(model.DecryptPurpose);

        model.DecryptedData = _protector.Unprotect(model.DataToDecrypt);
    }
    catch (Exception ex)
    {
        model.Exception = ex.ToString();
    }
    return RedirectToAction("Index", model);
}
```



ProtectDataController.cs

https://www.tn-data.se

Peeking inside the cookies

Peeking inside the cookies

ASP.NET Core issues several protected cookies

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie: .AspNetCore.Antiforgery.HfnB3ES-riM=CfDJ8BVbypiOrd4wz...
Set-Cookie: .AspNetCore.cookie=BQAAAAZjb29raWUBAAAABmNvb2tpZAAAOV...
Set-Cookie: .AspNetCore.Session=Mzk4OGM2YzYtMTU3MC0wN2FhLWkMWUyND...
…
```

How can we peek inside these cookies?

DEMO TIME!

Demonstrating data protection

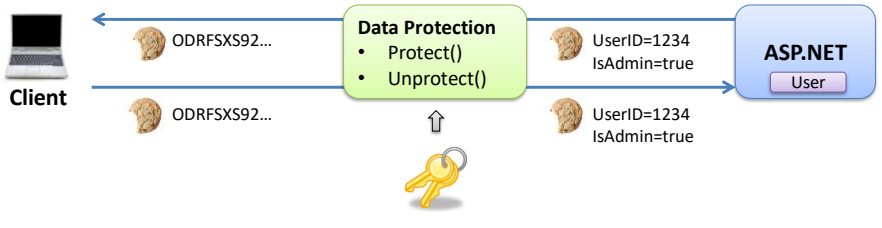
```
public class MyDataProtector : IDataProtector
{
    public IDataProtector CreateProtector(string purpose)
    {
        return new MyDataProtector();
    }

    public byte[] Protect(byte[] plaintext)
    {
        return plaintext;
    }

    public byte[] Unprotect(byte[] protectedData)
    {
        return protectedData;
    }
}
```

```
.AddCookie("cookie", o =>
{
    o.DataProtectionProvider = new MyDataProtector();
    //...
});
```

The Key ring

The Key ring

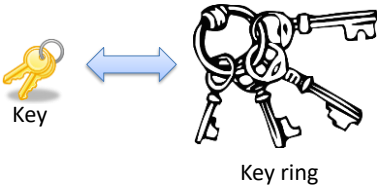
To encrypt data, we need to use an **encryption key**



Where should we store this key?

The Key ring

The key is stored in a **key ring**



Where should we store the **key ring**?

https://www.tn-data.se

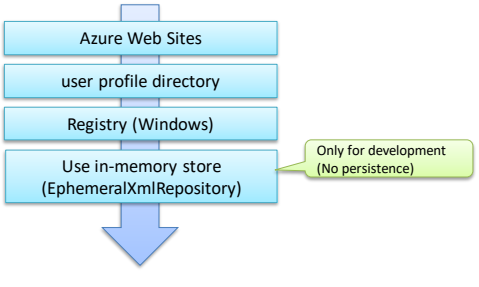
27

Where is the key ring located?

If we don't specify any location, like this:

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

Then it will look for it in the following places:



https://www.tn-data.se

28

DEMO TIME!

Locating the default location of the key ring in the logs

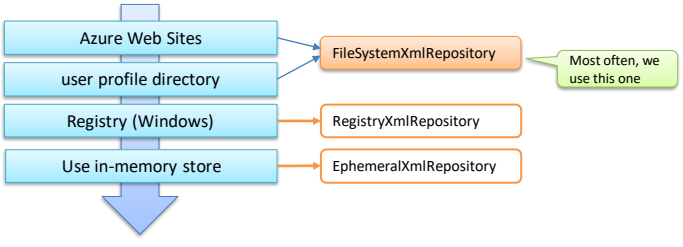
Storing the key ring

Storing the key ring

If we don't specify any persistence:

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

Then, it will use one of these persistence libraries



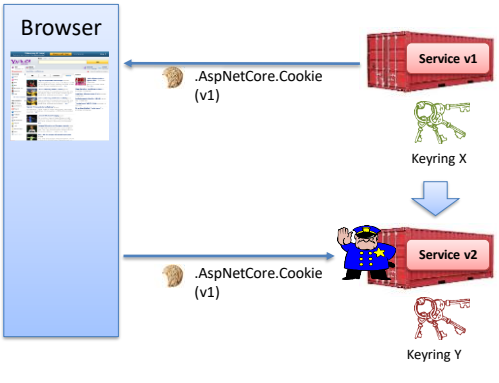
What is the problem with **FileSystemXmlRepository**?

https://www.tn-data.se

31

Storing the key ring

Data will be rejected if we lose it!



We need to ensure the key ring is **persisted**

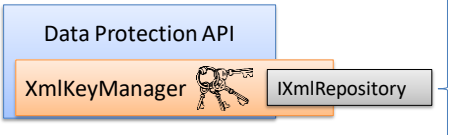
What options do we have?

https://www.tn-data.se

32

Data Protection API

The **key ring** can be stored in many places, including:



- AWS S3
- AWS Systems Manager
- Azure blob storage
- Dapper
- Entity Framework Core
- File System
- Google Cloud Storage
- In memory
- MongoDB
- NCache
- RavenDB
- Redis
- Registry
- Sqlite
- …

What does it take to implement an **IXmlRepository**?

More providers exists on NuGet
https://www.nuget.org/packages?q=AspNetCore.DataProtection

Data Protection API

Implementing one is simple; just implement this interface:

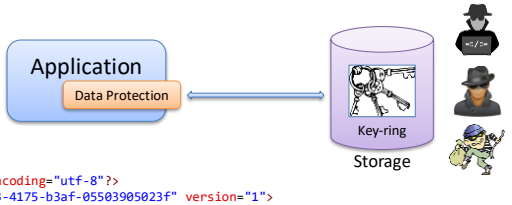
```
/// <summary>
/// The basic interface for storing and retrieving XML elements.
/// </summary>
public interface IXmlRepository
{
    /// <summary>
    /// Gets all top-level XML elements in the repository.
    /// </summary>
    /// <remarks>
    /// All top-level elements in the repository.
    /// </remarks>
    IReadOnlyCollection<XElement> GetAllElements();

    /// <summary>
    /// Adds a top-level XML element to the repository.
    /// </summary>
    /// <param name="element">The element to add.</param>
    /// <param name="friendlyName">An optional name to be associated with the XML element.
    /// For instance, if this repository stores XML files on disk, the friendly name may
    /// be used as part of the file name. Repository implementations are not required to
    /// observe this parameter even if it has been provided by the caller.</param>
    /// <remarks>
    /// The 'friendlyName' parameter must be unique if specified. For instance, it could
    /// be the id of the key being stored.
    /// </remarks>
    void StoreElement(XElement element, string friendlyName);
}
```

Protecting the key ring

Module #9

Protecting the key ring

The key ring is not protected when stored externally

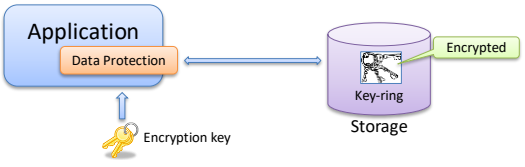



```xml
<?xml version="1.0" encoding="utf-8"?>
<key id="d7ce1df5-6bc3-4175-b3af-05503905023f" version="1">
  <creationDate>2024-04-21T12:38:01.0181934Z</creationDate>
  <activationDate>2024-04-21T12:38:01.0039009Z</activationDate>
  <expirationDate>2024-07-20T12:38:01.0039009Z</expirationDate>
  <descriptor deserializerType=".AuthenticatedEncryptorDescriptorDeserializer...">
    <descriptor>
      <encryption algorithm="AES_256_CBC" />
      <validation algorithm="HMACSHA256" />
      <encryptedSecret decryptorType="Microsoft.AspNetCore.DataProtection.XmlEncryption.NullXmlDecryptor..." >
        <unencryptedKey xmlns="">
          <!-- This key is not encrypted. -->
          <masterKey p6:requiresEncryption="true" xmlns:p6="http://schemas.asp.net/2015/03/dataProtection">
            <!-- Warning: the key below is in an unencrypted form. -->
            <value>AodL6W5URc3RNDa4GPS1mslGL7j8Hao8Ba90lLubs9LM6Bq1YRUs2qusIz+dLB/5XqG6uKPZGPSxGLoZ4eIw7A==</value>
          </masterKey>
        </unencryptedKey>
      </encryptedSecret>
    </descriptor>
  </descriptor>
</key>
```


How can we protect the keys at **rest**?

Protecting the key ring

There is an option to **encrypt** the keys at **rest**



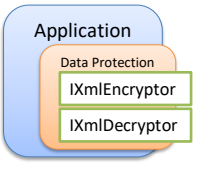

```xml
<?xml version="1.0" encoding="utf-8"?>
<key id="32cf330a-42bb-4820-a83c-1ebb5b9866fb" version="1">
  <creationDate>2024-04-21T12:41:23.2504886Z</creationDate>
  <activationDate>2024-04-21T12:41:23.2279982Z</activationDate>
  <expirationDate>2024-07-20T12:41:23.2279982Z</expirationDate>
  <descriptor deserializerType=".AuthenticatedEncryptorDescriptorDeserializer">
    <descriptor>
      <encryption algorithm="AES_256_CBC" />
      <validation algorithm="HMACSHA256" />
      <encryptedSecret decryptorType="Azure.Extensions.AspNetCore.DataProtection.Keys.AzureKeyVaultXmlDecryptor">
        <encryptedKey xmlns="">
          <!-- This key is encrypted with Azure Key Vault. -->
          <kid>https://dpapikeyvault.vault.azure.net/keys/DataProtectionKey/3fe072eca3a9497195f257228044f389</kid>
          <key>fdhKmO5aJ/orxOM3okH2/W8A4FFv2wBnW8lg/cEpe03MQpvDBKF55dGZH9...</key>
          <iv>Tbpr1jQbNvLRcmO8t7E+kw==</iv>
          <value>twWAS/AZBZujy1PkJ8Xq1rTiyNE6PoDqEXg0+QsgP2DlYd4WErukTQL/...<value>
        </encryptedKey>
      </encryptedSecret>
    </descriptor>
  </descriptor>
</key>
```

What are our options here?

https://www.tn-data.se

Protecting the key ring

There are plenty of options to choose from:



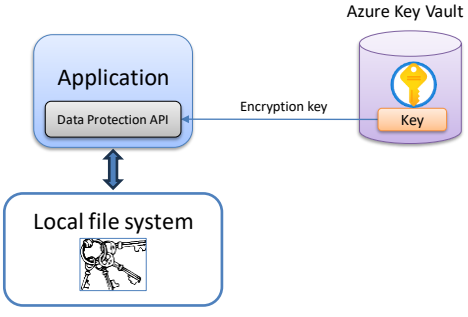
- **DpapiXml (Windows DPAPI)**
  - DpapiXmlEncryptor
  - DpapiXmlDecryptor
- **DpapiNG (Windows DPAPI:NG)**
  - DpapiNGXmlEncryptor
  - DpapiNGXmlDecryptor
- **EncryptedXml**
  - EncryptedXmlDecryptor
- **CertificateXml (using X.509 certificate)**
  - CertificateXmlEncryptor
- **NullXml**
  - NullXmlEncryptor
  - NullXmlDecryptor
- **InvalidEncryptor**
  - For read-only mode
- **Azure.Extensions.AspNetCore.DataProtection.Keys**

More can be found online and on GitHub

https://www.tn-data.se

Protecting the key ring

We can for example store the key in **Azure Key Vault**



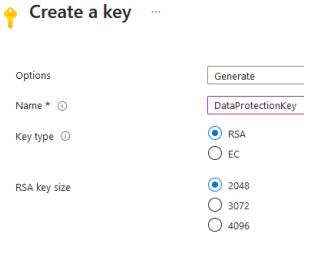
https://www.tn-data.se

DEMO TIME!

Encrypting the keys in the key ring

```
<PackageReference Include="Azure.Extensions.AspNetCore.DataProtection.Keys" Version="1.2.2" />
```

```
var keyUri = new
Uri("https://dpapikeyvault.vault.azure.net/keys/DataProtectionKey/2ebbb07bd1f3407e86ae5bdfc3e028fa");

builder.Services
    .AddDataProtection()
    .SetApplicationName("MyBusiness")
    .ProtectKeysWithAzureKeyVault(keyUri, new DefaultAzureCredential());
```




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

https://www.tn-data.se

Data Protection API Key Ring Debugger

More information

See my blog for more details: https://nestenius.se/

- Exploring what is inside the ASP.NET Core cookies
- Improving Security By Putting Your Cookies On A Diet
- Persisting the Data Protection Key Ring in Azure Key Vault
- Introducing the Data Protection API Key Ring Debugger
- …

Thanks for listening!

Win a 1-Hour Expert Presentation!

Elevate your workplace with fresh insights! 🌟

How to Enter:
Email me at: tore@tn-data.se

Prize:
2 lucky workplaces will receive a free 1-hour presentation!
(Within Skåne)

Draw Date:
Winners will be selected on 26-April 2024

Questions?
tore@tn-data.se