

# ASP.NET Core Identity

Implementing login/logout, Scaffolding



**SoftUni Team**  
Technical Trainers



**SoftUni**



Software University

<https://softuni.bg>

# Table of Contents

1. Authentication vs. Authorization
2. ASP.NET Core Identity
3. Scaffolding Identity

sli.do

**#csharp-web**



# **Authentication vs. Authorization**

# Authentication vs. Authorization

## ■ Authentication

- The process of verifying the identity of a user or computer
- Prerequisite for authorization
- Questions: **Who are you?** How you prove it?
- Credentials can be password, smart card, external token, etc.

## ■ Authorization

- The process of determining what a user is permitted to do on a computer or network
- Questions: **What are you allowed to do?** Can you see this page?
- You can't authorize a user before authenticating this user

# Authentication vs. Authorization (2)



## Authentication

Who you are



## Authorization

What you can do



**ASP.NET Core Identity**

- The **ASP.NET Core Identity** system
  - Authentication and authorization system for ASP.NET Core
  - Supports ASP.NET Core MVC, Pages, Web API (JWT), SignalR
  - Handles **Users, User Profiles, Login / Logout, Roles**, etc.
  - Handles cookie consent and GDPR
  - Supports external login providers
    - Facebook, Google, Twitter, etc.
  - Supports database, Azure, Active Directory, Windows Users, etc.



**Authentication**

Who you are



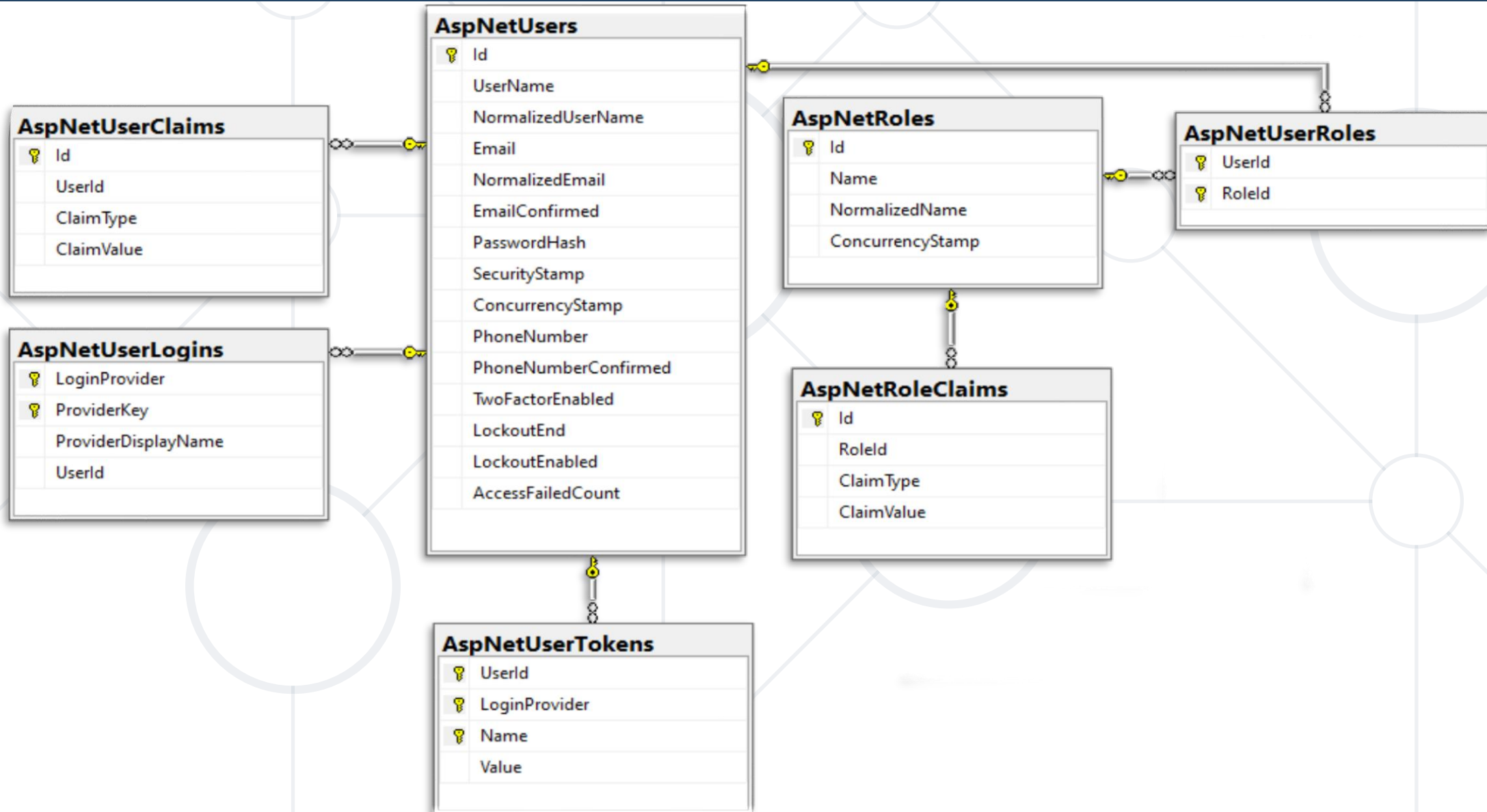
**Authorization**

What you can do



- Typically, the **ASP.NET Core** identity data is stored in relational database
  - Data is persisted using **Entity Framework Core**
  - You have some control over the internal database schema

# Internal Database Schema



- Setup **ASP.NET Identity**
  - Using the ASP.NET **project templates** from Visual Studio
    - And then customize it
  - **By hand**
    - Install NuGet packages, manual configuration, create EF mappings (models), view models, controllers, views, etc.
- Required NuGet package
  - **Microsoft.AspNetCore.Identity.EntityFrameworkCore**



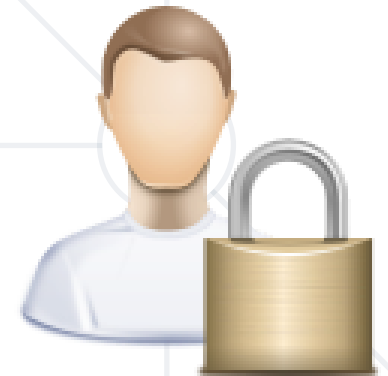
- **ApplicationDbContext.cs**

- Holds the EF data context
- Provides access to the application's data using model objects



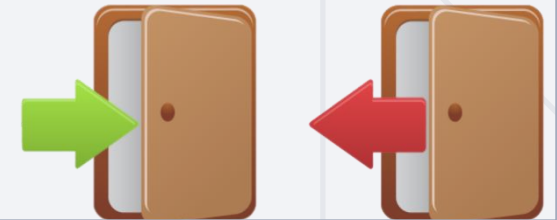
- **Program.cs**

- Can configure cookie-based (or JWT) authentication
- May enable external login (e.g. Facebook login)
- Can change default identity settings
  - More on those in the next course



- **Password settings** – can be defined in **Program.cs**

```
builder.Services.AddDefaultIdentity<IdentityUser>(options =>
{
    // Password, lockout, emails, etc.
    options.SignIn.RequireConfirmedAccount = false;
    options.Password.RequireDigit = false;
})
.AddEntityFrameworkStores<ApplicationDbContext>();
```



- Use the `[Authorize]` and `[AllowAnonymous]` attributes to configure **Authorized / Anonymous access** for **Controller / Action**

```
[Authorize]
public class AccountController : Controller
{
    // GET: /Account/Login (anonymous)
    [AllowAnonymous]
    public async Task<IActionResult> Login(string returnUrl) { ... }

    // POST: /Account/LogOff (for logged-in users only)
    [HttpPost]
    public async Task<IActionResult> Logout() { ... }
}
```

# Check the Currently Logged-In User

```
// GET: /Account/Roles (for logged-in users only)
[Authorize]
public ActionResult Roles()
{
    var currentUser = await userManager.GetUserAsync(this.User);
    var roles = await userManager.GetRolesAsync(currentUser);
    ...
}
```

```
// GET: /Account/Data (for logged-in users only)
[Authorize]
public ActionResult Data()
{
    var currentUser = await userManager.GetUserAsync(this.User);
    var currentUserUsername = await userManager.GetUserNameAsync(currentUser);
    var currentUserId = await userManager.GetUserIdAsync(currentUser);
    ...
}
```

- **UserManager<TUser>** - APIs for managing users in a persistence store

Category		
AddClaimsAsync(...)	FindByEmailAsync(...)	GenerateChangePhoneNumberTokenAsync(...)
AddToRoleAsync(...)	FindByIdAsync(...)	GenerateEmailConfirmationTokenAsync(...)
IsInRoleAsync(...)	FindByNameAsync(...)	GeneratePasswordResetTokenAsync(...)
GetUserId(...)	GetClaimsAsync(...)	GetAuthenticationTokenAsync(...)
ChangeEmailAsync(...)	GetEmailAsync(...)	IsEmailConfirmedAsync(...)
ConfirmEmailAsync(...)	GetRolesAsync(...)	CreateSecurityTokenAsync(...)
CreateAsync(...)	GetUserAsync(...)	ResetPasswordAsync(...)
DeleteAsync(...)	CheckPasswordAsync(...)	RemoveFromRoleAsync(...)
Dispose(...)	UpdatePassword(...)	RemoveClaimsAsync(...)

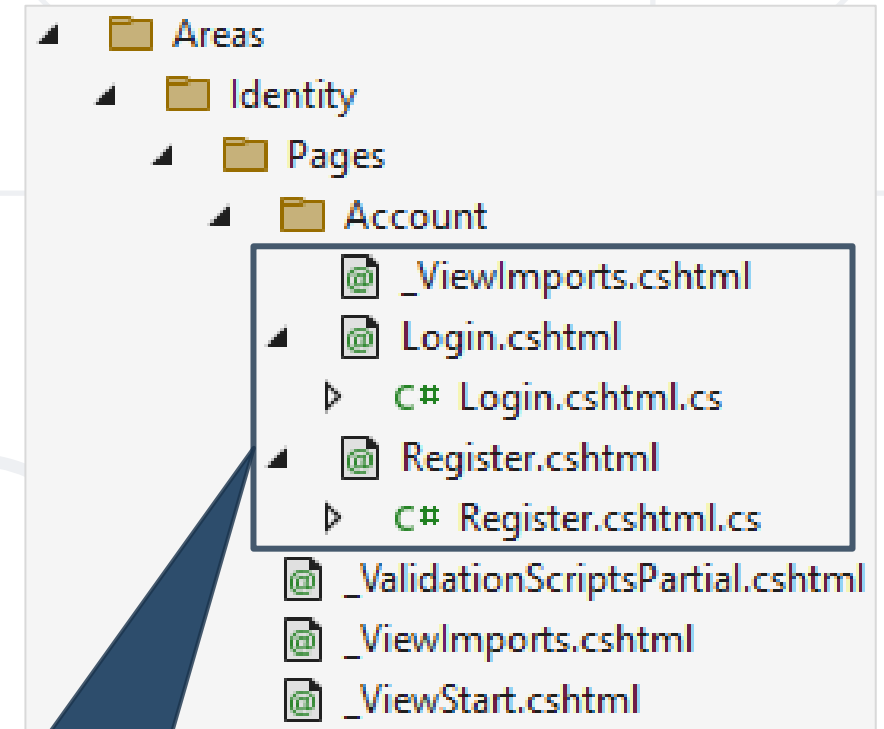




**Scaffolding Identity**

# Scaffolding ASP.NET Core Identity

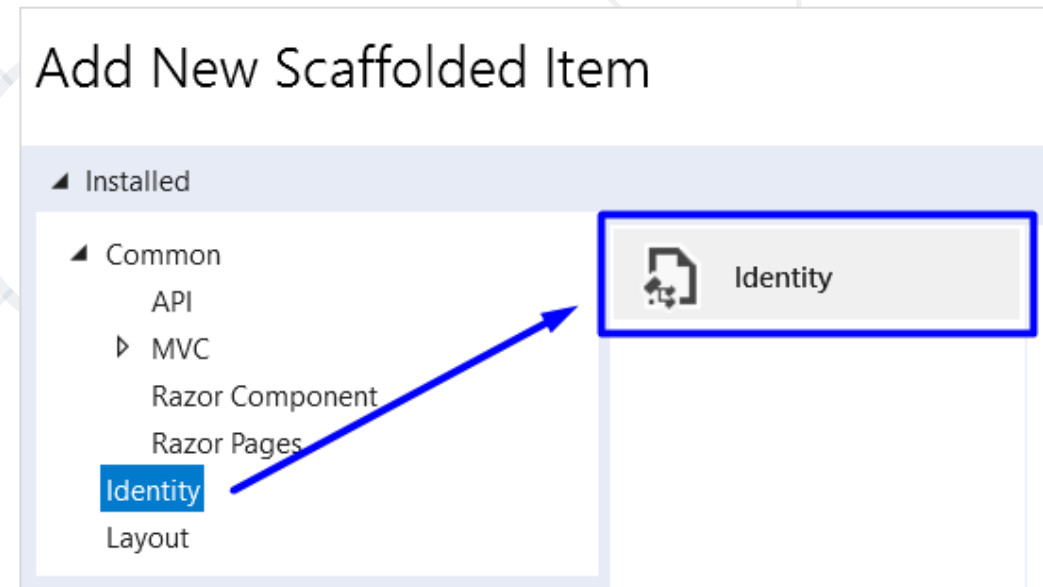
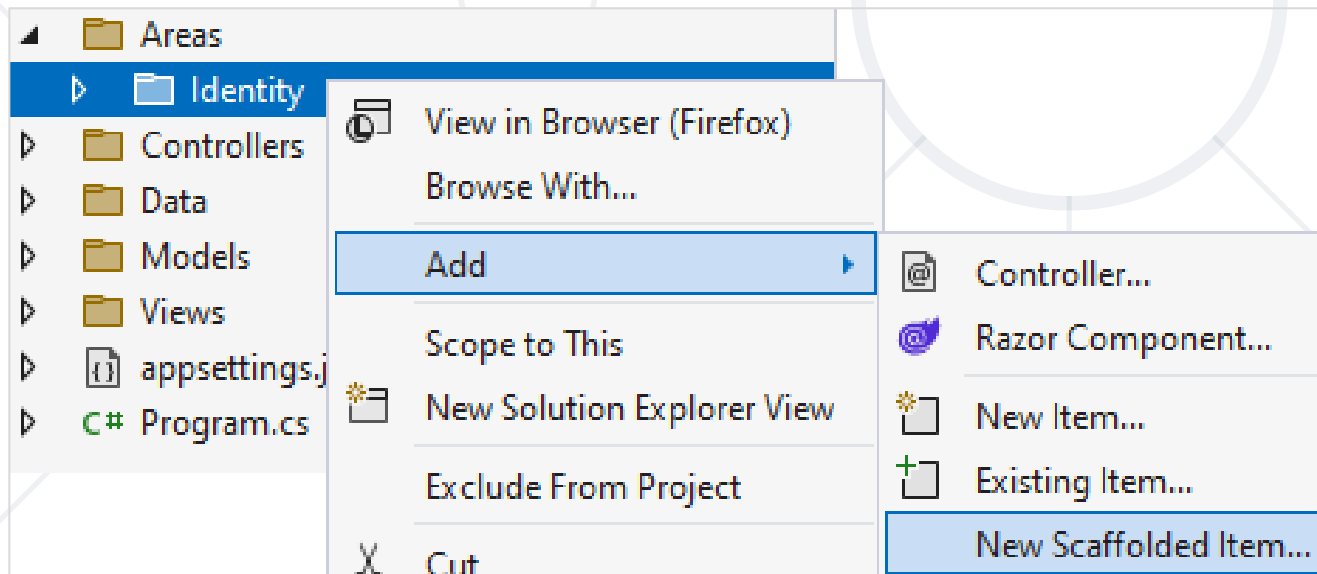
- Since **ASP.NET Core 2.2**, **Identity** is provided as a **Razor Class Library**
- The **scaffolder** can be configured to generate source code
  - If you need to modify the code and change the behavior
- Most of the necessary code is generated by the **scaffolder**



Scaffolded  
**Account.Register**  
and **Account.Login**

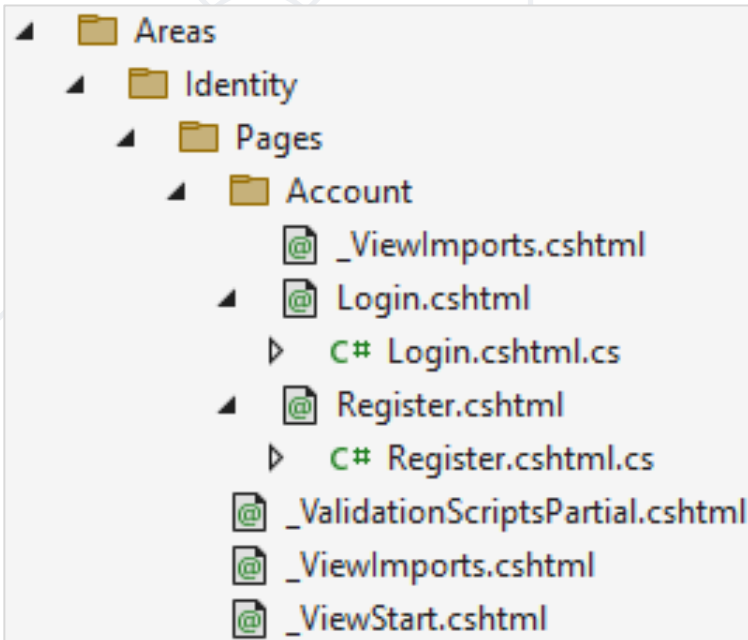
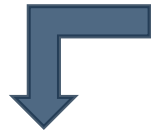
# Scaffolding ASP.NET Core Identity in VS

- Scaffold Identity pages by adding a **new scaffolded identity item**



# Scaffolding ASP.NET Core Identity in VS (2)

- **Scaffolded pages** will be part of the **`"/Areas/Identity"`** folder



**Add Identity**

Select an existing layout page, or specify a new one:

(Leave empty if it is set in a Razor \_viewstart file)

☐ Override all files

Choose files to override

<input type="checkbox"/> Account\StatusMessage	<input type="checkbox"/> Account\AccessDenied	<input type="checkbox"/> Account\ConfirmEmail
<input type="checkbox"/> Account\ConfirmEmailChange	<input type="checkbox"/> Account\ExternalLogin	<input type="checkbox"/> Account\ForgotPassword
<input type="checkbox"/> Account\ForgotPasswordConfirmation	<input type="checkbox"/> Account\Lockout	<input checked="" type="checkbox"/> Account\Login
<input type="checkbox"/> Account>LoginWith2fa	<input type="checkbox"/> Account>LoginWithRecoveryCode	<input type="checkbox"/> Account\Logout
<input type="checkbox"/> Account\Manage\Layout	<input type="checkbox"/> Account\Manage\ManageNav	<input type="checkbox"/> Account\Manage\StatusMessage
<input type="checkbox"/> Account\Manage\ChangePassword	<input type="checkbox"/> Account\Manage\DeletePersonalData	<input type="checkbox"/> Account\Manage\Disable2fa
<input type="checkbox"/> Account\Manage\DownloadPersonalData	<input type="checkbox"/> Account\Manage\Email	<input type="checkbox"/> Account\Manage\EnableAuthenticator
<input type="checkbox"/> Account\Manage\ExternalLogins	<input type="checkbox"/> Account\Manage\GenerateRecoveryCodes	<input type="checkbox"/> Account\Manage\Index
<input type="checkbox"/> Account\Manage\PersonalData	<input type="checkbox"/> Account\Manage\ResetAuthenticator	<input type="checkbox"/> Account\Manage\SetPassword
<input type="checkbox"/> Account\Manage\ShowRecoveryCodes	<input type="checkbox"/> Account\Manage\TwoFactorAuthentication	<input checked="" type="checkbox"/> Account\Register
<input type="checkbox"/> Account\RegisterConfirmation	<input type="checkbox"/> Account\ResendEmailConfirmation	<input type="checkbox"/> Account\ResetPassword
<input type="checkbox"/> Account\ResetPasswordConfirmation		

Data context class:

☐ Use SQLite instead of SQL Server

User class:

**Add** **Cancel**

Select the  
**layout page**

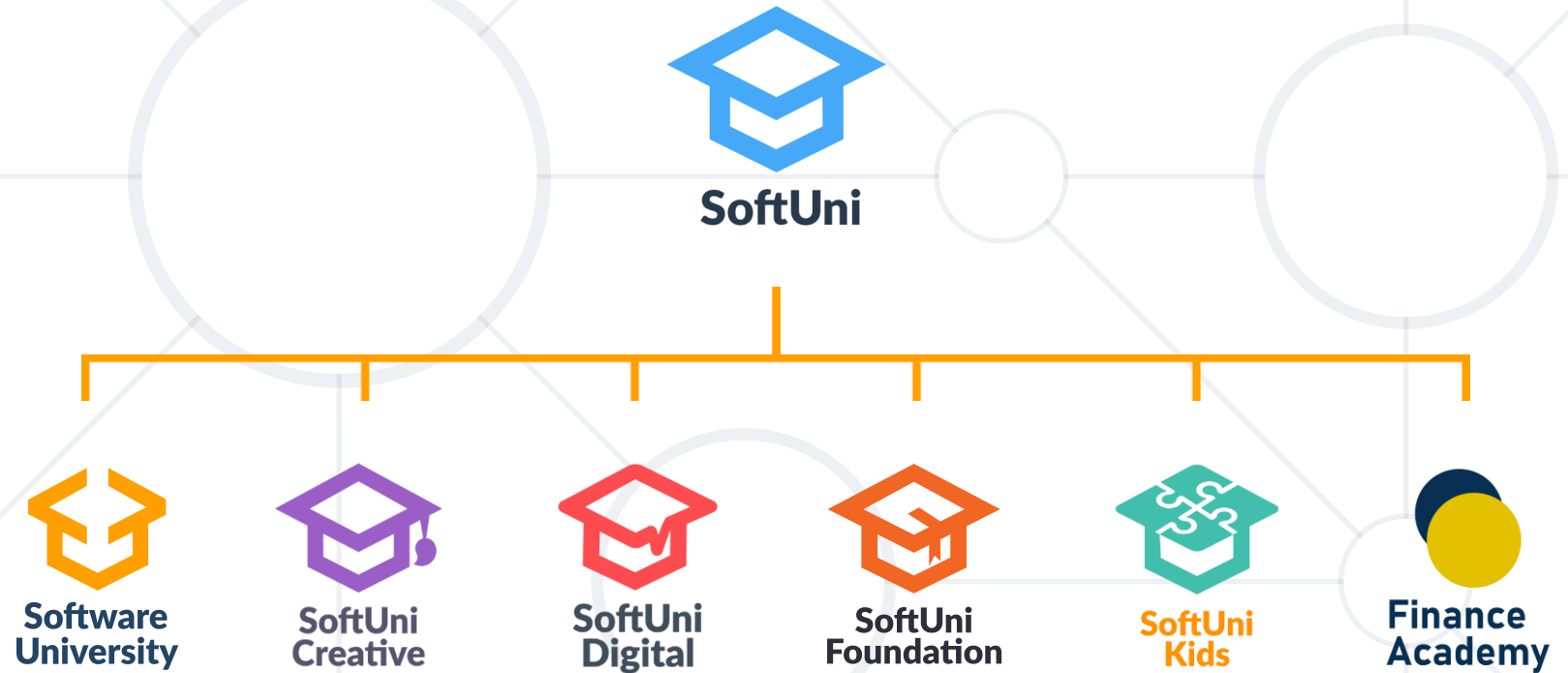
Select  
**pages to be scaffolded**

Select the  
**db context**

1. Authentication vs. Authorization
2. **ASP.NET Core Identity**
3. **Scaffolding** Identity



# Questions?



# SoftUni Diamond Partners

**SUPER  
HOSTING  
.BG**



**Coca-Cola HBC  
Bulgaria**



**POKERSTARS**  
POKER | CASINO | SPORTS  
a Flutter International brand

**INDEAVR**  
Serving the high achievers



**AMBITIONED**

 **DRAFT  
KINGS**



**SOFTWARE  
GROUP**

createX



**Postbank**  
Решения за твоето утре

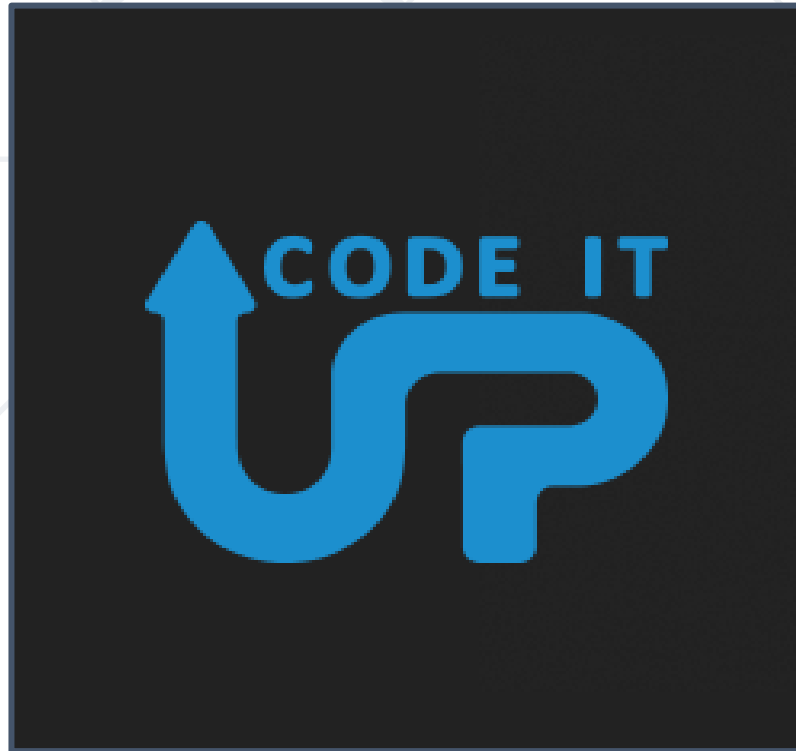


**BOSCH**

**DXC**  
TECHNOLOGY



**SmartIT**





- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg>
- © Software University – <https://softuni.bg>

