

Adding Login Capabilities



Gill Cleeren

CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be



Overview



Exploring ASP.NET Core Identity

Adding authentication to the site

Enabling authorization



Exploring ASP.NET Core Identity





ASP.NET Identity

- Membership system
- Authentication and authorization
- Supports external providers
- SQL Server support built-in
- Since ASP.NET Core 2.1
 - Scaffolding support



Adding ASP.NET Identity

- IdentityDbContext
- Configuration changes
 - Password length
 - Cookies
 - User options
- Scaffolding

Configuration Options

```
services.Configure<IdentityOptions>(options =>
{
    options.Password.RequireDigit = true;
    options.Password.RequiredLength = 8;
    options.Password.RequireNonAlphanumeric = true;
    options.User.RequireUniqueEmail = true;
});
```



Demo

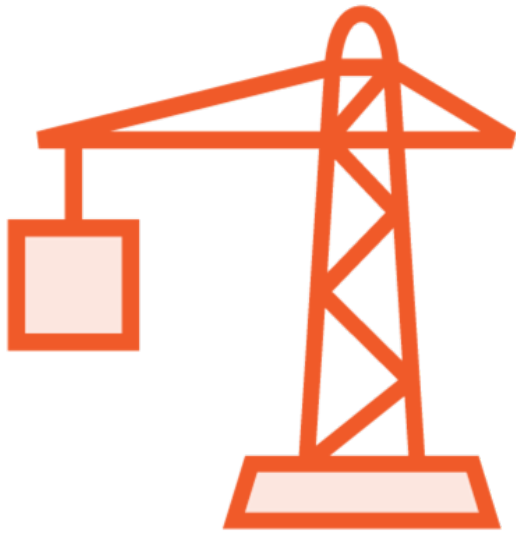


Adding ASP.NET Identity



Adding Authentication to the Site





Scaffolding

- Supported in ASP.NET Core 2.1 and 3.0
- Manual approach still works
- Based on Razor Class Library



Side Note: Razor Class Library



Added with ASP.NET Core 2.1

Reuse functionality

- Views, pages, controllers, view components...

Can be overridden


Razor Class Library (RCL)


×


Create a new ASP.NET Core Web Application


.NET Core


ASP.NET Core 2.1


 **API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

 **Web Application**
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content.

 **Web Application (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

 **Razor Class Library**
A project template for creating a Razor class library.

 **Angular**
A project template for creating an ASP.NET Core application with Angular.

 **React.js**

[Get additional project templates](#)

Back

Create

Authentication

No Authentication
[Change](#)

Advanced

☒ Configure for HTTPS
☐ Enable Docker Support
(Requires Docker Desktop)

Linux



Allowing Users to Log In

Use built-in RCL

Scaffold items

Layout changes



Important Classes in Identity

UserManager<IdentityUser>

SignInManager<IdentityUser>



Demo



**Scaffolding Login and Register
functionality**

Layout changes



Enabling Authorization



[Authorize]

```
public class OrderController : Controller  
{  
}
```

Authorizing Users




```
public class OrderController : Controller
{
    [Authorize]
    public IActionResult Checkout()
    {
    }
}
```

Action Level



```
[Authorize(Roles = "Administrator")]  
public class OrderController : Controller  
{  
}
```

Combining with Roles



Demo



Authorizing users to place orders



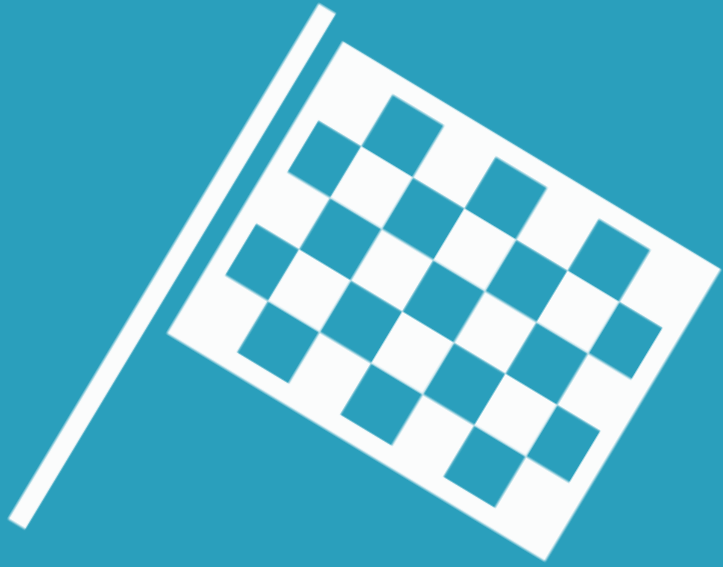
Summary



ASP.NET Identity works for

- authentication
- authorization





Congratulations
on finishing this course!

