Cookie-based Authentication with ASP.NET Core Identity

Kevin Dockx

ARCHITECT

@KevinDockx https://www.kevindockx.com

Coming Up

Positioning ASP.NET Core Identity

Replacing our current approach with ASP.NET Core Identity

Potential issues

Positioning ASP.NET Core Identity

Membership system that support user interface login functionality

- Provides views, classes and methods we can use to login, logout, create and edit users and store them in a database

We’re focusing on how to integrate it with Blazor

- ASP.NET Core Identity Deep Dive (Scott Brady)

Demo

Scaffolding ASP.NET Core Identity into a Blazor application

Demo

Authenticating with ASP.NET Core Identity (part 1)

Demo

Including an AntiForgeryToken for POST requests

Demo

Authenticating with ASP.NET Core Identity (part 2)

Issues with Cookie-based Authentication

We still don’t have a good solution for authentication/authorization at level of our API

- We should use tokens for that instead of cookies

A client application needs proof of authentication, it doesn’t need access to the user database

- Use an identity provider

Summary

ASP.NET Core Identity is a membership system that support user interface login functionality

- Select “authentication with individual user accounts” when creating a new project
- Scaffold it into existing projects

Summary

Cookie authentication isn’t a good fit for APIs

A client application needs proof of authentication, it doesn’t need access to the user database