

Token-based Authentication with ASP.NET Core Identity



Kevin Dockx

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up



**Token-based authentication with
ASP.NET Core Identity**



Integrating Token-based Security with ASP.NET Core Identity

Issues with our previous approach

- The cookie is great for protecting the Blazor application, but not for the API
- Our Blazor application shouldn't be responsible for user-related tasks



Integrating Token-based Security with ASP.NET Core Identity

We'll integrate ASP.NET Core Identity's membership system and database at level of the IDP

- This allow our Blazor application and API to keep relying on token-based security



Integrating Token-based Security with ASP.NET Core Identity

We'll integrate ASP.NET Core Identity's user management screens at level of the IDP

- This is common, but not obligatory

The main takeaway is that this functionality must be centralized



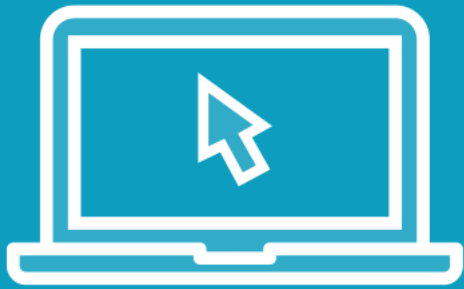
Demo



Integrating token-based security with ASP.NET Core Identity



Demo



Adding ASP.NET Core Identity's user management screens



Summary



By integrating ASP.NET Core Identity's membership system and database at level of the IDP

- We were able to reuse the built-in membership functionality
- We allowed our Blazor application and API to keep relying on token-based security



Summary



By integrating ASP.NET Core Identity's user management screens at level of the IDP, we were able to centralize that functionality

- It can now be reused by other client applications that rely on the same user database / IDP

