educación IT
CODE YOUR FUTURE

Programación Web .NET Core

Módulo 10

Personalizar proveedores con ASP.NET Core

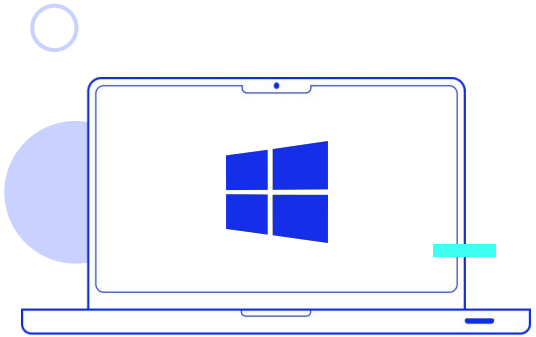
Personalizar proveedores con ASP.NET Core

ASP.NET Core Identity, listo para usar, utiliza **Entity Framework** para manejar la autenticación y maneja la autenticación de extremo a extremo dentro del ámbito de aplicación. Esto puede estar bien en muchos casos pero, de vez en cuando, se necesitarán opciones adicionales de negocio. Los requisitos pueden dictar la necesidad de utilizar la infraestructura interna del directorio activo para iniciar sesión, o es posible que se desee permitir que usuarios de **Frameworks** alternativos usen la aplicación. Incluso, que no se utilice **SQL Server** para almacenamiento de datos.

Debes tener en cuenta que estas exigencias requieren cambios extensos en su aplicación y deben trabajarse y decidirse de antemano.

Autenticación de Windows

Al crear una aplicación diseñada para usarse en un entorno de intranet, se debe considerar el uso de ***Autenticación de Windows***. Al configurar la autenticación de Windows, los usuarios podrán iniciar sesión en la aplicación utilizando sus credenciales de Windows. Al utilizar la autenticación de Windows, el ***directorio activo*** se usará para determinar las credenciales de usuario y permisos, permitiendo que los usuarios sean administrados y creados por administradores internos. Esto ayuda a crear aplicaciones y elimina la necesidad de administrar usuarios manualmente.

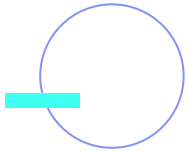


Es posible que se desee utilizar la autenticación de Windows cuando:

- La aplicación está diseñada para ser utilizada dentro de una organización.
- Cuando se desee que la creación de usuarios se limite y gestione manualmente.
- Para evitar usuarios indeseados en la aplicación.
- Cuando la aplicación presenta información sensible para una empresa.

No se debe utilizar la autenticación de Windows cuando:

- Es necesario crear aplicaciones accesibles desde Internet.
- Cuando se desee que los usuarios puedan registrarse y crear sus propias cuentas en la aplicación.
- Cuando la empresa no está administrada por **Windows Active Directory**.



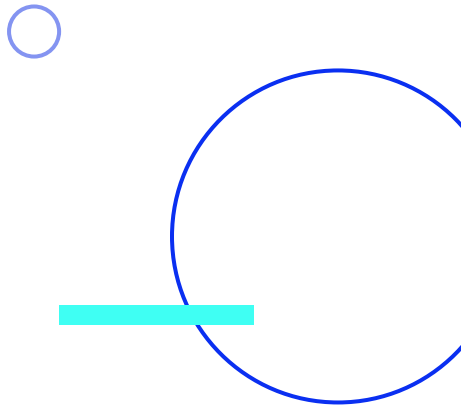
Utilización de proveedores externos

De forma predeterminada, en una aplicación ASP.NET Core, todos los inicios de sesión se manejan a través de la propia aplicación. Sin embargo, el proceso de registro puede ser tedioso y muchos usuarios ya poseen varios usuarios diferentes para sitios web diferentes.

Para simplificar, es posible implementar el inicio de sesión mediante **proveedores externos**. Por ejemplo, los que detallamos a la derecha:

- **Microsoft:** como lo utilizan una variedad de productos de Microsoft, como Microsoft **Azure** o **Hotmail**.
- **Facebook:** como se usa en **Facebook** y su aplicación móvil.
- **Twitter:** como se usa en **Twitter** y su aplicación móvil.
- **Google:** como se usa para aplicaciones de Google como **Gmail** y **Google Maps**.
- Otras opciones.

Implementar un soporte para un proveedor externo, permite que tus usuarios inicien sesión a través de un servicio en el que ya tienen una cuenta. Como resultado de una conexión exitosa, tu aplicación recibirá un **token** de éxito que luego se puede usar para obtener información adicional sobre el usuario. No existe una forma única y uniforme de configurar proveedores, ya que cada proveedor utiliza diferentes protocolos y lógica. Siempre se debe considerar el costo/beneficio de agregar ese proveedor.



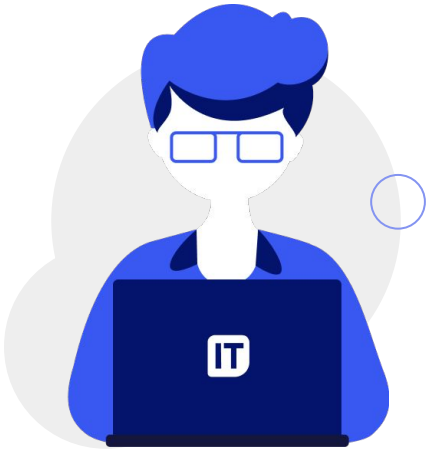
Uso de proveedores de almacenamiento alternativos

De forma predeterminada, ASP.NET Core MVC utiliza **Entity Framework Core** para administrar usuarios en la aplicación. Esto puede ser conveniente para configurar soluciones de almacenamiento simples para la aplicación. Sin embargo, no siempre es una opción viable. Muchas veces durante el proceso de desarrollo, es posible que se deba utilizar una opción de almacenamiento diferente a **SQL Server** o incluso utilizar datos en un formato diferente.

Ejemplos de escenarios que incluyen, pero no se limitan a:

- Almacenamiento de tablas de **Azure**.
- Bases de datos con una estructura de datos diferente como **MongoDB** o **Redis**.
- Utilizar sistemas de acceso a datos distintos de **EntityFramework Core**.

Para respaldar a estos proveedores de almacenamiento, deberás escribir un nuevo proveedor personalizado. El código específico depende del proveedor elegido. Para implementarlo, deberás crear una lógica de almacenamiento de identidad alternativa, así como una capa de acceso a datos alternativa.



educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!