Session 13

.NET Core Token Authentication in Web Applications

Session Overview

- Describe token-based authentication
- Explain the process to validate tokens in ASP .NET core
- Describe the automatic authorization of metadata
- Explain symmetric and asymmetric keys
- Explain the process to generate tokens for authentication in ASP .NET Core

Overview

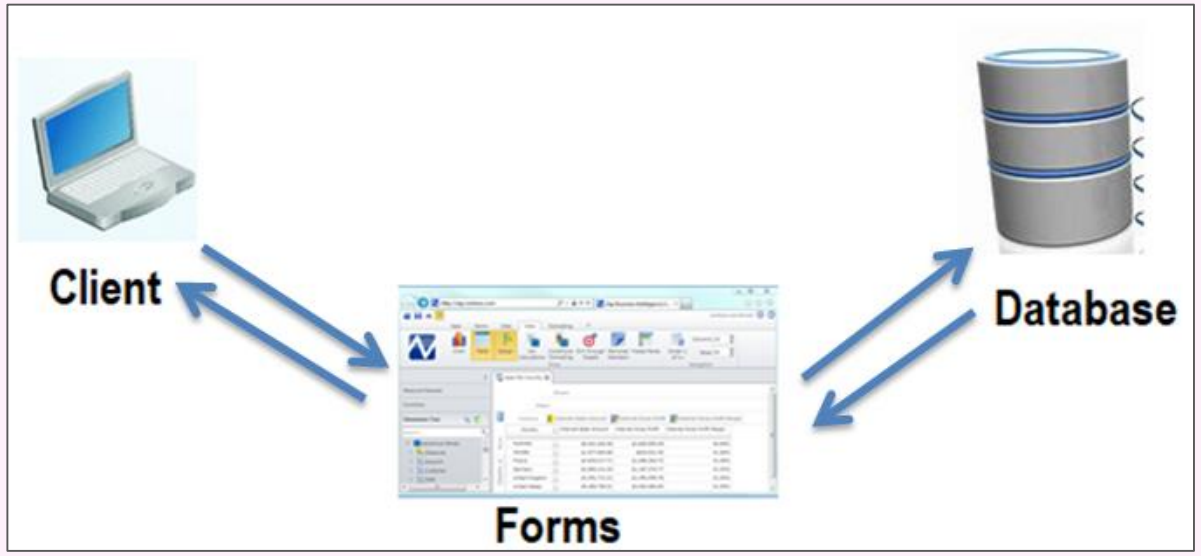

Figure 13.1: Storing Data in Sessions

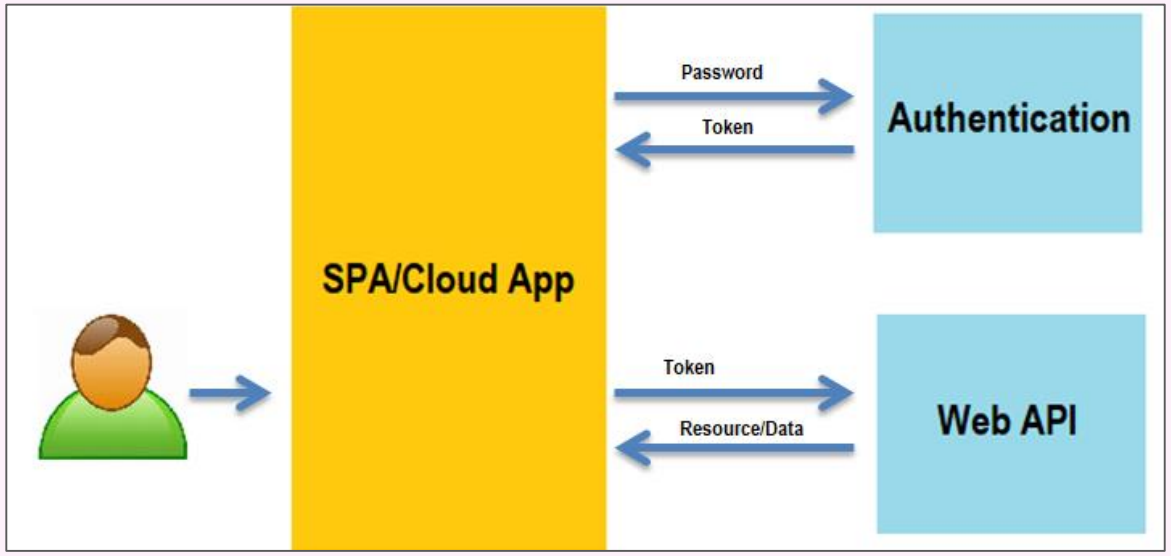

Figure 13.2: Data Storage in Could-Based Environment and SPAs

Token-Based Authentication

- Provides an additional encrypted security process in the authentication process.
- Is very secure and difficult to be hacked.
- Is used with APIs in mobile and SPAs.
- Can be performed through JSON Web Token (JWT) which is an open-source technology and can we used with any backend technology, such as ASP.NET, Python, or Java.

© Aptech Web Development with ASP.NET MVC and ASP.NET CORE/Session 13

Validating Tokens in ASP.NET Core

ValidateAudience	To validate the recipient of the token.
ValidAudience	Is the value of the audience that is used for validation.
RequireExpirationTime	To check if the token has any expiry time.
ValidateLifetime	To check expiry of the token.
ClockSkew	To validate time.
IssuerSigningKeys	To validate signature of the token.
ValidateIssuer	To validate the issuer of the token, the server that generates the token.

Symmetric and Asymmetric Encryption

Symmetric Encryption

- Uses same key for encryption and decryption.
- Is quicker.
- Can handle vast volumes of text, streams, or files.

Asymmetric Encryption

- Uses different keys for encryption and decryption.
- Is slower.
- Can handle only small quantities of data.

Generating Tokens for Authentication in ASP.NET Core

Main methods for obtaining an authorization server:

Using a cloud service such as Azure

An authorization server generates tokens for OpenID Connect or OAuth 2.0. Authorization server is also used to establish access policies. Issuer URI and signing key for tokens is used to segregate the security domains.

Creating or configuring own server

A custom authorization server can be built with unique OAuth 2.0 contexts, assertions, and access management. It provides API authorization within each authorization server.

Summary

- The most preferred way of transferring session storage to the cloud is using REST design.
- The cloud-based environment allows users to use Web service APIs directly from the user interface.
- Token-based authentication provides an additional encrypted security process in the authentication process.
- Token-based authentication is secured and difficult to be hacked. This kind of authentication is used with APIs in mobile and SPAs.
- Tokens must be validated after it has been created. However, tokens can be validated only as a string.
- Metadata can be retrieved from the authorization server using JwtBearer middleware.
- Depending on whether a symmetric or asymmetric key is used, the keys to be used to sign in the token will be submitted.
- Symmetric encryption uses the same key. However, asymmetric encryption uses different keys for encryption and decryption.
- The two main methods for obtaining an authorization server are using a cloud service such as Azure and creating or configuring own server.