🚀 30 TRICKY INTERVIEW QUESTIONS ON AUTHENTICATION & AUTHORIZATION IN .NET CORE WEB API – WITH DETAILED ANSWERS 💡

Are you preparing for a .NET Core interview or looking to master secure Web API development? Here's your go-to guide with **30 advanced Authentication & Authorization questions**, perfect for cracking interviews or leveling up your skills. ✅

🔐 AUTHENTICATION VS AUTHORIZATION

1. **What is the difference between Authentication and Authorization?**
   - **Authentication** verifies *who* you are.
   - **Authorization** defines *what* you can access.
   - Example: Logging in = Authentication; Viewing dashboard = Authorization.

🔑 JWT TOKEN-BASED AUTHENTICATION

2. **What is JWT and why is it used in Web API?**
   - JWT (JSON Web Token) is a compact, URL-safe token used to transfer claims between two parties. It's widely used for **stateless authentication**.
3. **How do you generate JWT in .NET Core?**
   - Use System.IdentityModel.Tokens.Jwt.
   - Configure TokenValidationParameters and generate token using JwtSecurityTokenHandler.
4. **What are claims in JWT?**
   - Claims are key-value pairs that represent user identity (e.g., Name, Email, Role).
5. **What is the purpose of the 'Issuer' and 'Audience' in JWT?**
   - Issuer: Who created the token.
   - Audience: Who the token is intended for.

🧩 ROLE-BASED AUTHORIZATION

6. **How is role-based authorization implemented in .NET Core?**
   - Decorate actions or controllers using [Authorize(Roles = "Admin")].
7. **Can a user have multiple roles in JWT?**
   - Yes. Use multiple "role" claims or an array of roles in the JWT payload.

8. **How do you secure endpoints for multiple roles?**

[Authorize(Roles = "Admin,Manager")]

POLICY-BASED AUTHORIZATION

9. **What is policy-based authorization?**

    ○ Instead of static roles, policies allow custom logic via requirements and handlers.

10. **How do you define a custom policy?**

- Use services.AddAuthorization(options => {...}) and register requirements.

11. **How do you implement a custom AuthorizationHandler?**

- Inherit from AuthorizationHandler<TRequirement> and override HandleRequirementAsync.

TOKEN REFRESH & EXPIRY

12. **What happens when a JWT token expires?**

- The client gets a 401 Unauthorized. You can implement a **refresh token** strategy.

13. **How do you implement Refresh Tokens?**

- Store refresh tokens server-side. On expiry, send a new access token using the refresh token.

OAUTH2 & OPENID CONNECT

14. **What is the difference between OAuth2 and OpenID Connect?**

- OAuth2 is for authorization.
- OpenID Connect is built on OAuth2 and adds authentication features.

15. **How to implement Google/Facebook login in .NET Core?**

- Use AddAuthentication().AddGoogle() or AddFacebook() in Startup.cs.

IDENTITYSERVER & EXTERNAL PROVIDERS

16. **What is IdentityServer?**

- A powerful framework for implementing OAuth2 and OpenID Connect in .NET Core.

17. **How do you secure an API using IdentityServer4?**

- Configure IdentityServer, generate tokens, and validate them in the Web API using middleware.

ASP.NET CORE IDENTITY

18. **What is ASP.NET Core Identity?**

- A full-featured membership system for managing users, passwords, roles, and claims.

19. **How do you override password rules in Identity?**

- Configure PasswordOptions in IdentityOptions during service configuration.

MIDDLEWARE & TOKEN VALIDATION

20. **Where should you validate JWT tokens in the middleware pipeline?**

- After UseRouting() but before UseEndpoints().

21. **How can you access the currently logged-in user?**

- Use HttpContext.User.Identity.Name or access claims from User.Claims.

ADVANCED & TRICKY SCENARIOS

22. **Can you invalidate a JWT token before it expires?**

- JWTs are stateless, so you must implement token revocation via a server-side blacklist.

23. **How to prevent token replay attacks?**

- Use short-lived tokens with refresh strategy, and track usage on the server.

24. **Is HTTPS required for JWT?**

- Yes. JWTs contain sensitive info and must be transmitted over HTTPS.

25. **How to secure Swagger endpoints?**

- Use [Authorize] on controllers and configure Swagger to accept Bearer tokens.

26. **What's the risk of putting roles in the JWT?**

- If the token is compromised, all role info is exposed. Always encrypt tokens and use HTTPS.

MULTI-TENANCY & CLAIMS

27. **How to implement multi-tenancy authorization?**

- Include tenant info in claims and validate it in custom handlers or filters.

28. **How do you manage per-tenant roles?**

- Use custom claims like TenantId and role mapping logic based on tenant.

⚙ MISCELLANEOUS

29. **Difference between [Authorize], [AllowAnonymous], and [Authorize(Roles = "...")]?**

- [Authorize]: Requires authentication.
- [AllowAnonymous]: Skips auth.
- [Authorize(Roles = "...")]: Requires specific role.

30. **How to log authorization failures?**

- Hook into OnChallenge or OnForbidden events in JWT bearer options.

BONUS TIPS

- Always secure APIs using HTTPS.
- Rotate keys and secrets regularly.
- Prefer short-lived tokens with refresh strategy.
- Validate tokens strictly (issuer, audience, lifetime, signing key).

💬 LET ME KNOW IN THE COMMENTS WHICH QUESTION WAS MOST HELPFUL!
👍 LIKE & 🔁 REPOST TO HELP OTHERS ACE THEIR .NET CORE INTERVIEWS!