AZ-204 - 3. Implement Azure security -> 3.1 Implement user authentication and authorization -> 2.2.1 Authenticate and authorize users by using the Microsoft Identity platform

1. What is the Microsoft Identity Platform?
2. How do you register an application with Microsoft Entra ID (formerly Azure AD)?
3. What is the difference between single-tenant and multi-tenant apps?
4. What authentication flows are supported in the Microsoft Identity Platform?
5. How do you implement authentication using MSAL?
6. How do you configure permissions (scopes) and consent?
7. How do you acquire and validate access tokens?
8. How do you secure an API using the Microsoft Identity Platform?
9. How do you configure redirect URIs and reply URLs?
10. What is the difference between delegated and application permissions?

---

1. What is the Microsoft Identity Platform?

A Microsoft authentication system that provides OAuth 2.0 and OpenID Connect protocols for authenticating users and securing APIs. It integrates with Microsoft Entra ID (Azure AD).

---

2. How do you register an application with Microsoft Entra ID?

Use Azure Portal → Entra ID → App registrations → New registration.
Set a name, supported account types, and redirect URI. Save the Application (client) ID.

---

3. What is the difference between single-tenant and multi-tenant apps?

- Single-tenant: Only users in one Entra ID tenant can access the app.
- Multi-tenant: Users in any Entra ID tenant can authenticate.

---

4. What authentication flows are supported in the Microsoft Identity Platform?

- Authorization Code (interactive user login)
- Client Credentials (daemon apps)
- Device Code (devices without browser)
- ROPC (username/password; not recommended)
- On-Behalf-Of (service-to-service delegation)

---

5. How do you implement authentication using MSAL?

Use the Microsoft Authentication Library (MSAL) to acquire tokens:

```
var result = await app.AcquireTokenInteractive(scopes).ExecuteAsync();
```

MSAL handles caching, token renewal, and multiple flows.

---

6. How do you configure permissions (scopes) and consent?

Define scopes in the app registration under **Expose an API**.

- Admins or users must **consent** to scopes (e.g., user.read).
- API permissions tab controls delegated vs. application scopes.

7. How do you acquire and validate access tokens?

Use MSAL to acquire tokens (e.g., AcquireTokenInteractive, AcquireTokenForClient).

Validate tokens in the API using middleware (e.g., ASP.NET JwtBearerOptions) and Microsoft identity metadata endpoint.

---

8. How do you secure an API using the Microsoft Identity Platform?

- Register the API as an application.
- Define scopes under "Expose an API".
- Protect routes using [Authorize] and validate tokens using middleware (AddAuthentication().AddJwtBearer()).

---

9. How do you configure redirect URIs and reply URLs?

Set in Azure Portal → App registration → Authentication.

- Must match what's used in your app exactly.
- Used during OAuth flows to redirect users back after authentication.

---

10. What is the difference between delegated and application permissions?

- **Delegated**: Act on behalf of a user. Used with signed-in users.
- **Application**: Act as the app itself. Used in background services (e.g., daemons).