

Service-Lebenszyklen in Dependency Injection: Singleton, Scoped und Transient

Die Begriffe Scoped, Singleton und Transient beziehen sich auf Lebenszyklen von Diensten in der Abhängigkeitsinjektion (Dependency Injection, DI), besonders in Frameworks wie ASP.NET Core.

Uebersicht:

1. Singleton:

- Eine einzige Instanz pro Anwendung.
- Registrierung: `services.AddSingleton<IMyService, MyService>();`
- Ideal fuer zustandslose, thread-sichere Komponenten.

2. Scoped:

- Eine Instanz pro HTTP-Anfrage.
- Registrierung: `services.AddScoped<IMyService, MyService>();`
- Ideal fuer services wie DbContext.

3. Transient:

- Neue Instanz bei jedem Aufruf.
- Registrierung: `services.AddTransient<IMyService, MyService>();`
- Ideal fuer zustandslose, leichtgewichtige Services.

Vergleich:

- Singleton: global geteilt
- Scoped: geteilt innerhalb einer Anfrage
- Transient: nie geteilt

Empfehlung:

- Singleton: fuer Konfiguration, Logging
- Scoped: fuer Benutzer-/Request-bezogene Services
- Transient: fuer stateless Utility-Services