Isha Fatima • 2nd
Full Stack Software Engineer | .NET Core & Angular Specialist | Blazorise, MVC, Web API | Solvi...
1h •

Follow

Dependency Injection Best Practices: **Scoped vs Singleton vs Transient**!

Choosing the right DI lifetime is crucial for performance & stability in .NET apps.

Let's break it down:

Transient – New Instance Every Time

🔷 Use Case: Lightweight, stateless services.

🔷 Example: Repository, Email sender, Logging.

🔷Too many instances = increased memory usage if used excessively.

Scoped – One Instance Per Request

🔷 Use Case: Services tied to an HTTP request.

🔷 Example: Database context (EF Core), Business logic services.

🔷Don't inject Scoped services into Singleton services! It may cause unexpected issues.

Singleton – One Instance for the App's Lifetime

🔷 Use Case: Heavy, shared resources that shouldn't be re-created.

🔷 Example: Caching, Configuration, Logging, Static Data.

🔷Memory leaks if holding unnecessary state. Avoid injecting Scoped/Transient services here

Avoid Service Lifetime Mismatches

Good: Singleton -> Uses other Singletons

Bad: Singleton -> Uses Scoped/Transient (may cause unintended behavior)

#dotnet #dependencyinjection #csharp #di #backend #microservices #netdevelopers

11 2 comments 1 repost

Like Comment Repost Send

Anton Martyniuk • Following 24m
Microsoft MVP | Helping 30K+ Software Engineers Improve .NET Skills and Craft Better Softw...

Note: you can use IServiceProvider to create a scope that can be used to create scoped service inside a Singleton class