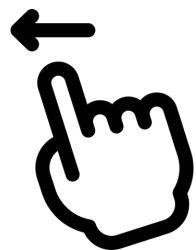


.net Tip



DEPENDENCY INJECTION (DI) LIFETIMES IN .NET

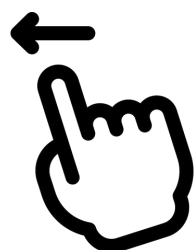


SINGLETON

A single instance of the dependency is created and shared across the entire application. This is the longest-lived lifetime and is useful for stateful services that maintain internal state and need to be shared across the application.

A small icon of a terminal window with three colored circles (red, yellow, green) in the top left corner.

```
services.AddSingleton<IService, MyService>();
```



SCOPED

A new instance of the dependency is created once per scope. A scope can be thought of as a unit of work, such as an HTTP request in a web application. All dependencies that are created within the same scope will share the same instance of the scoped dependency.



```
services.AddScoped<IService, MyService>();
```

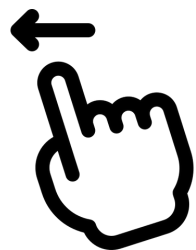


TRANSIENT

A new instance of the dependency is created each time it is requested. This is the shortest-lived lifetime and is useful for stateless services that do not maintain any internal state.



```
services.AddTransient<IService, MyService>();
```



THANKS FOR READY
SHARE YOUR THOUGHTS

FOLLOW ME FOR MORE SHARING

