.NET Dependency Injection
Logging
Generic Host

> Microsoft.Extensions.*

.NET Dependency Injection

Dependencies


```
Add (using microsoft.Exten
AddScoped
AddScoped<>
AddSingleton
AddSingleton<>
AddTransient<>
AddTransient
AddKeyedScoped
AddKeyedScoped<>
AddKeyedSingleton
```

} = ▷ ≥ 8.0

DI Container

```
services.AddScoped<>();
services.AddTransient<>();
services.AddSingleton<>();
```

Transient -> Jede Anfrage, mit einer neuen, frischen, eigenständigen Instanz beantwortet

Singleton -> Jede Anfrage, bekommt immer die selbe Instanz

Scoped -> Innerhalb eines Scopes wird immer die selbe Instanz zurückgegeben

```
var loggerFactory = LoggerFactory
    .Create( configure: x :ILoggingBuilder =>
    {
        x.AddConsole().AddJsonConsole();
        x.AddProvider(new MyLoggerProvider());
    });

ILogger<Program> logger = loggerFactory.CreateLogger<Program>();
logger.LogInformation( message: "Juhu {Datum} {UserId}", params args: DateTime.Now, 42);
```

HostedServices

```
await new HostBuilder()
    .ConfigureServices(services :IServiceCollection =>
    {
        // DI Registrierungen
    })
    .ConfigureLogging(loggingBuilder =>
    {
        loggingBuilder.AddConsole();
    }) // IHostBuilder
    .RunConsoleAsync(); // Task
```

APP

Hosted