

.NET Dependency Injection
Logging
Generic Host

> Microsoft.Extensions.*

.NET Dependency Injection

Dependencies

new B
A
Class

B
Class

ng

Mic

Dep

2-P

-->

-->

Add (using Microsoft.Extensions.DependencyInjection)

AddScoped

AddScoped<>

AddSingleton

AddSingleton<>

AddTransient<>

AddTransient

AddKeyedScoped

AddKeyedScoped<>

AddKeyedSingleton

=> 8.0

DI Container

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

services.AddScoped<>();

services.AddTransient<>();

services.AddSingleton<>();

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

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

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

Hosted Services

APP
Hosted