



CONFIGURATION

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

KONFIGURATIONSDATEIEN

- **appsettings.json**
 - Hauptkonfigurationsdatei für allgemeine Einstellungen
- **appsettings.Development.json**
 - Umgebungspezifische Konfigurationsdatei für die Entwicklungsumgebung
- **launchSettings.json**
 - Definiert Startprofile und Umgebungsvariablen (nur in Visual Studio)

APPSETTINGS.JSON

- Enthält allgemeine, umgebungsunabhängige Einstellungen
- Der Connection String für eine SQLite-Datenbank kann hier platziert werden unter dem Abschnitt ConnectionStrings

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=C:\\Users\\Vince\\Desktop\\test.db"
  }
}
```

APPSETTINGS.DEVELOPMENT.JSON

- Spezifisch für die Entwicklungsumgebung
- Überschreibt oder ergänzt die Einstellungen aus appsettings.json

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Debug",  
      "Microsoft.AspNetCore": "Warning"  
    }  
  }  
}
```

LAUNCHSETTINGS.JSON DATEI

- Definiert Startprofile und Umgebungsvariablen für die Anwendung in Visual Studio
- Hier kann die Umgebungsvariable ASPNETCORE_ENVIRONMENT auf Development gesetzt werden, um die Entwicklungsumgebung zu aktivieren

```
{
  "$schema": "http://json.schemastore.org/launchsettings.json",
  "https": {
    "commandName": "Project",
    "dotnetRunMessages": true,
    "launchBrowser": true,
    "applicationUrl": "https://localhost:7045;http://localhost:5218",
    "environmentVariables": {
      "ASPNETCORE_ENVIRONMENT": "Development"
    }
  }
}
```

ICONFIGURATION

- **IConfiguration Interface**
- Dient zum Abrufen von Konfigurationswerten in ASP.NET Core Anwendungen
- Ermöglicht den Zugriff auf Konfigurationsdaten aus verschiedenen Quellen wie JSON-Dateien, Umgebungsvariablen, Befehlszeilenargumenten usw.
- Konfigurationswerte können über das IConfiguration Interface abgerufen werden

```
var scope = app.Services.CreateScope();  
var configuration = scope.ServiceProvider.GetRequiredService<IConfiguration>();  
  
Debug.WriteLine(configuration["ASPNETCORE_ENVIRONMENT"]);  
Debug.WriteLine(configuration["AllowedHosts"]);  
Debug.WriteLine(configuration.GetRequiredSection("Logging").GetSection("LogLevel").GetSection("Default").Value);  
// Oder  
Debug.WriteLine(configuration["Logging:LogLevel:Default"]);  
Debug.WriteLine(configuration.GetConnectionString("DefaultConnection"));
```

ICONFIGURATION

- Auch via Dependency Injection nutzbar

```
public class ApplicationDbContext : DbContext
{
    private readonly IConfiguration _configuration;

    public ApplicationDbContext(IConfiguration configuration)
    {
        _configuration = configuration;
    }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlite(_configuration.GetConnectionString("DefaultConnection"));
    }
}
```