

## Lab 1

Dauer 10 Minuten

Ziel

Das Erstellen einer ASP.NET-Anwendung über die Kommandozeile erfordert die Verwendung des .NET Command Line Interface (CLI), das als "dotnet" bezeichnet wird. Hier sind die Schritte, um eine einfache ASP.NET-Anwendung zu erstellen:

Schritt 1: Installation von .NET Core SDK (falls nicht vorhanden)

Stellen Sie sicher, dass Sie das .NET Core SDK auf Ihrem System installiert haben. Sie können es von der offiziellen .NET-Website herunterladen und installieren: <https://dotnet.microsoft.com/download>

Schritt 2: Erstellen eines neuen ASP.NET-Projekts

1. Öffnen Sie eine Kommandozeile (z.B. die Eingabeaufforderung unter Windows oder das Terminal unter macOS/Linux).
2. Navigieren Sie zu dem Verzeichnis, in dem Sie Ihr Projekt erstellen möchten.
3. Führen Sie den folgenden Befehl aus, um ein neues ASP.NET-Projekt zu erstellen:

```
dotnet new webapp -o MeineAspNetApp
```

Hier erstellen wir ein neues Webprojekt mit dem Namen "MeineAspNetApp". Sie können den Namen entsprechend anpassen.

Schritt 3: Navigieren in das Projektverzeichnis

Wechseln Sie in das Verzeichnis des neu erstellten Projekts:

```
cd MeineAspNetApp
```

Schritt 4: Starten der Anwendung

Führen Sie den folgenden Befehl aus, um die ASP.NET-Anwendung zu starten:

```
dotnet run
```

Dieser Befehl kompiliert das Projekt und startet den eingebauten Entwicklungsserver. Sie werden eine Meldung sehen, die angibt, auf welcher Adresse die Anwendung läuft (normalerweise `http://localhost:5000`).

#### Schritt 5: Öffnen der Anwendung im Webbrowser

Öffnen Sie Ihren Webbrowser und geben Sie die URL ein, auf der die Anwendung läuft (z.B. `http://localhost:5000`). Sie sollten die Standard-ASP.NET-Startseite sehen.

Hinweis: Nachdem die Anwendung gestartet ist, können Sie den Code im Verzeichnis "MeineAspNetApp" bearbeiten. Die Hauptdatei für Ihre Anwendung ist normalerweise "programm.cs", in der Sie Routen, Dienste und Konfigurationen festlegen können.

Dies sind die grundlegenden Schritte, um eine ASP.NET-Anwendung über die Kommandozeile zu erstellen und auszuführen. Sie können dann Schritt für Schritt weitere Funktionen und Seiten hinzufügen, um Ihre Anwendung zu erweitern.

Natürlich, hier ist ein einfaches Beispiel, wie Sie Ihre ASP.NET-Anwendung erweitern können, indem Sie eine neue Seite hinzufügen, die eine Liste von Produkten anzeigt:

#### Schritt 1: Hinzufügen einer neuen Seite für die Produktliste

1. Öffnen Sie das Projektverzeichnis Ihrer ASP.NET-Anwendung und wechseln sie in das Pages Verzeichnis

2. Führen Sie den folgenden Befehl aus, um eine neue Razor Page hinzuzufügen:

```
dotnet new page -n ProduktListe -na MeineAspNetApp
```

Hier wird eine neue Razor Page mit dem Namen "ProduktListe" im Namespace "MeineAspNetApp" erstellt.

## Schritt 2: Anpassen der neuen Seite

Öffnen Sie die neu erstellte Razor Page "ProduktListe.cshtml" im "Pages" Verzeichnis. Hier können Sie den generierten Code anpassen, um die Produktliste anzuzeigen. Hier ist ein einfaches Beispiel:

```
@page
```

```
@model MeineAspNetApp.Pages.ProduktListeModel
```

```
<h1>Produktliste</h1>
```

```
<ul>
```

```
    @foreach (var produkt in Model.Produkte)
```

```
    {
```

```
        <li>@produkt.Name - @produkt.Preis</li>
```

```
    }
```

```
</ul>
```

## Schritt 3: Erstellen des Modells für die Seite

Erstellen Sie im gleichen Verzeichnis wie die Razor Page eine Datei mit dem Namen "ProduktListe.cshtml.cs" und definieren Sie das Modell für die Seite:

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Threading.Tasks;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
using Microsoft.AspNetCore.Mvc.RazorPages;
```

```
namespace MeineAspNetApp.Pages
```

```

{
    public class ProduktListeModel : PageModel
    {
        public List<Produkt> Produkte { get; set; }

        public void OnGet()
        {
            // Beispielprodukte hinzufügen
            Produkte = new List<Produkt>
            {
                new Produkt { Name = "Produkt 1", Preis = 10.99 },
                new Produkt { Name = "Produkt 2", Preis = 19.99 },
                new Produkt { Name = "Produkt 3", Preis = 5.49 }
            };
        }
    }
}

public class Produkt
{
    public string Name { get; set; }
    public double Preis { get; set; }
}
}

```

#### Schritt 4: Anzeigen der Seite im Browser

Starten Sie die Anwendung mit ``dotnet run`` und öffnen Sie im Browser die URL `http://localhost:5000/ProduktListe`. Sie sollten die Produktliste angezeigt bekommen.

Dies ist nur ein einfaches Beispiel dafür, wie Sie eine neue Seite zu Ihrer ASP.NET-Anwendung hinzufügen können. Sie können weitere Funktionen hinzufügen, Datenbankverbindungen einbauen und das Layout anpassen, um Ihre Anwendung zu erweitern.