

Schritt-für-Schritt-Anleitung zur Erstellung einer Angular-Anwendung

Diese Anleitung beschreibt die Schritte zur Installation von Node.js, dem Angular CLI und zur Erstellung und Ausführung einer neuen Angular-Anwendung auf verschiedenen Betriebssystemen.

Schritt 1: Node.js installieren

Windows:

1. Gehen Sie zur Node.js-Website (<https://nodejs.org/>).
2. Laden Sie den Windows-Installer herunter (empfohlen ist die LTS-Version).
3. Öffnen Sie die heruntergeladene Installationsdatei.
4. Folgen Sie den Anweisungen des Installationsprogramms und wählen Sie die Option "Add to PATH".
5. Öffnen Sie die Eingabeaufforderung (CMD) oder PowerShell und geben Sie ``node -v`` und ``npm -v`` ein, um die Installation zu überprüfen.

macOS:

1. Öffnen Sie das Terminal.
2. Installieren Sie Homebrew, falls es noch nicht installiert ist:
`/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
3. Installieren Sie Node.js mit Homebrew:
`brew install node`
4. Überprüfen Sie die Installation:
`node -v`
`npm -v`

Linux (Ubuntu/Debian):

1. Öffnen Sie das Terminal.
2. Fügen Sie das NodeSource-Repository hinzu und installieren Sie Node.js:
`curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -`
`sudo apt-get install -y nodejs`
3. Installieren Sie Build-Tools (optional, aber empfohlen):
`sudo apt-get install -y build-essential`
4. Überprüfen Sie die Installation:
`node -v`
`npm -v`

Schritt 2: Angular CLI installieren

Öffnen Sie das Terminal (oder die Eingabeaufforderung auf Windows) und führen Sie den folgenden Befehl aus:

```
npm install -g @angular/cli
```

Dies installiert das Angular Command Line Interface (CLI) global auf Ihrem Rechner.

Schritt 3: Eine neue Angular-Anwendung erstellen

Navigieren Sie im Terminal zu dem Verzeichnis, in dem Sie Ihre Angular-Anwendung erstellen möchten. Dann führen Sie den folgenden Befehl aus:

```
ng new meine-angular-anwendung
```

Beantworten Sie die Fragen, die das Angular CLI stellt (z.B. ob Sie Angular Routing hinzufügen möchten und welchen Stil Sie verwenden möchten).

Schritt 4: Die Anwendung starten

Navigieren Sie in das neu erstellte Verzeichnis Ihrer Angular-Anwendung:

```
cd meine-angular-anwendung
```

Starten Sie die Anwendung mit dem folgenden Befehl:

```
ng serve
```

Öffnen Sie Ihren Webbrowser und navigieren Sie zu <http://localhost:4200>, um Ihre neue Angular-Anwendung in Aktion zu sehen.

Schritt 5: Die Anwendung schöner starten

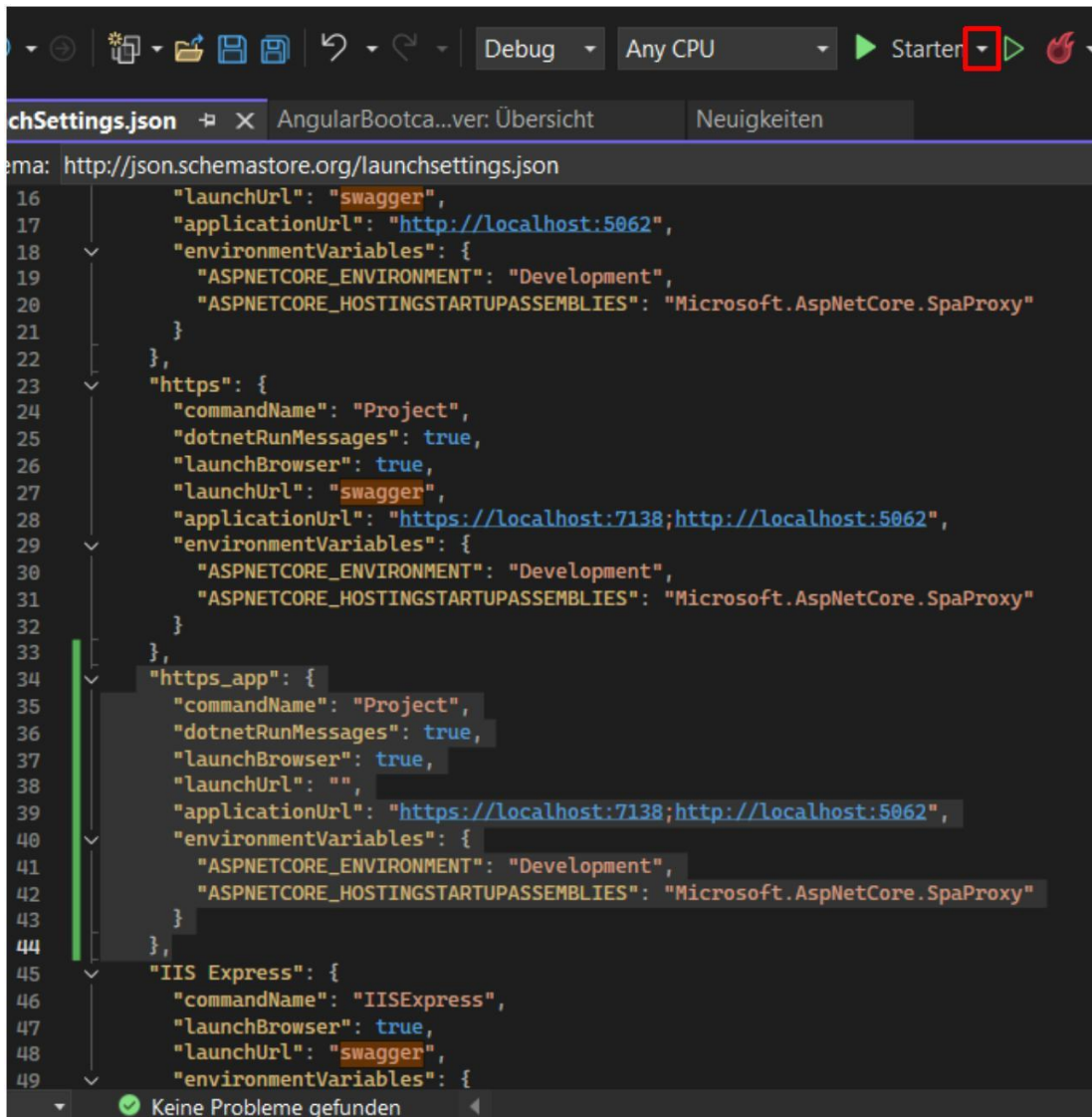
Um nicht wie im Schritt 4 manuell navigieren zu müssen und den localhost:4200 aufrufen zu müssen können Sie bestimmte Einstellungen ändern.

Gehen Sie dafür in die `launchSettings.json` Ihrer Angular Applikation.

Sie können folgenden Code unter dem Codestück `"https"`: indem `"launchURL": "swagger"` einfügen:

```
"https_app": {
  "commandName": "Project",
  "dotnetRunMessages": true,
  "launchBrowser": true,
  "launchUrl": "",
  "applicationUrl": "https://localhost:7138;http://localhost:5062",
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development",
    "ASPNETCORE_HOSTINGSTARTUPASSEMBLIES": "Microsoft.AspNetCore.SpaProxy"
  }
},
```

Danach:



```
16  "launchUrl": "swagger",
17  "applicationUrl": "http://localhost:5062",
18  "environmentVariables": {
19    "ASPNETCORE_ENVIRONMENT": "Development",
20    "ASPNETCORE_HOSTINGSTARTUPASSEMBLIES": "Microsoft.AspNetCore.SpaProxy"
21  },
22  },
23  "https": {
24    "commandName": "Project",
25    "dotnetRunMessages": true,
26    "launchBrowser": true,
27    "launchUrl": "swagger",
28    "applicationUrl": "https://localhost:7138;http://localhost:5062",
29    "environmentVariables": {
30      "ASPNETCORE_ENVIRONMENT": "Development",
31      "ASPNETCORE_HOSTINGSTARTUPASSEMBLIES": "Microsoft.AspNetCore.SpaProxy"
32    }
33  },
34  "https_app": {
35    "commandName": "Project",
36    "dotnetRunMessages": true,
37    "launchBrowser": true,
38    "launchUrl": "",
39    "applicationUrl": "https://localhost:7138;http://localhost:5062",
40    "environmentVariables": {
41      "ASPNETCORE_ENVIRONMENT": "Development",
42      "ASPNETCORE_HOSTINGSTARTUPASSEMBLIES": "Microsoft.AspNetCore.SpaProxy"
43    }
44  },
45  "IIS Express": {
46    "commandName": "IISExpress",
47    "launchBrowser": true,
48    "launchUrl": "swagger",
49    "environmentVariables": {
```

Keine Probleme gefunden

