

02 - Angularprojekt erstellen

Neues Projekt

```
ng new <project-name>
```

Option	Beschreibung	Default-Wert
routing	Routing-Konfiguration	false
prefix	Selektor-Präfix	app
dry-run	ohne Ausführung	false
--create-application	Anwendung hinzufügen	true

```
ng new <project-name> --create-application=false
```

Projektstruktur

Aufbau

```
Projekt
├── documentation
├── src
│   ├── app
│   │   ├── modules      ---auch Unterteilung nach weiteren Modulen---
│   │   │   ├── components
│   │   │   ├── layouts
│   │   │   ├── pages
│   │   │   ├── modules-routing.module.ts
│   │   │   └── modules.module.ts
│   │   ├── core
│   │   │   ├── authentication
│   │   │   ├── footer
│   │   │   ├── guards
│   │   │   ├── models
│   │   │   ├── navbar
│   │   │   ├── services
│   │   │   ├── sidebar
│   │   │   └── core.module.ts
│   │   ├── config
│   │   ├── shared
│   │   │   ├── components
│   │   │   └── directives
```

```

├── pipes
├── validators
├── shared.module.ts
├── app-routing.module.ts
├── app.component.ts
├── app.module.ts
├── assets
│   ├── configs
│   ├── images
│   └── favicon

```

Komponenten generieren

Angular Anwendungen bestehen aus Komponenten.

```

ng generate component <component-name>
ng g c 'component-name'

```

Es wird daraufhin die Komponente mit den Dateien `*.component.css`, `*.component.html`, `*.component.ts` und `*.component.spec.ts` erstellt.

Option	Beschreibung	Default-Wert
flat	Erstellung direkt im aktuellen Ordner	false
inline-template	Der HTML-Template wird direkt in der <code>*.ts</code> -Datei erzeugt.	false
inline-style	Das Styling wird ebenfalls direkt in der <code>*.ts</code> -Datei integriert.	false
skipTests	Testdatei nicht erzeugen	false

```
ng g c 'name' --inline-style=true
```

Weitere Generatoren

Befehl	Beschreibung
<code>ng g c 'name'</code>	Erstellen einer Komponente
<code>ng g s 'name'</code>	Erstellen einer Service-Klasse
<code>ng g m 'name'</code> , <code>ng g m 'name' --routing=true</code>	Erstellen eines Modules mit einem Router
<code>ng g class 'name'</code>	Erstellen einer Klasse
<code>ng g interface 'name'</code>	Erstellen eines Interface
<code>ng g guard 'name'</code>	Erstellen eines Router-Guard

<code>ng g application 'name'</code>	Erstellt ein zusätzliches Projekt (Multi-Project)
<code>ng g library 'name'</code>	Erzeugt eine Library

Erweiterte Projektstruktur (Cordova, Tailwind CSS und Angular Material)

```
# create angular project
ng new <project_name> --create-application=false

# add cordova
cordova create <project_name> de.block-code.app "CordovaTestApp"

# create application
cd <project_name>
ng g application <app_name>

# install tailwind by npm
cd projects/<app_name>
npm install -D tailwindcss postcss@latest autoprefixer@latest
npx tailwindcss init

# add angular material
ng add @angular/material --project=<app_name>
```

Angular Material kann alternativ auch per `npm` hinzugefügt werden. Dazu ist das Package in die Datei `package.json` hinzuzufügen:

```
"dependencies": {
  ..
  "@angular/material": "17.3.0"
},
```

Anschließend wird in der `angular.json` folgende Veränderung vorgenommen:

```
{
  "projects": {
    "application": {
      "architecture": {
        "build": {
          "options": {
            "outputPath": "www",
            ...
          },
          ...
        },
        ...
      },
      ...
    },
    ...
  },
  ...
},
```

```
    ...  
  }
```

Durch diese Einstellung wird das Ergebnis des Build in den Ordner `./www` abgelegt.

In der `tailwind.config.js` muss dann noch der Pfad angepasst werden:

```
module.exports = {  
  content: [  
    // './src/**/*.html,ts,css,scss,sass,less,styl',  
    './projects/**/*.html,ts,css,scss,sass,less,styl'  
  ],  
  ...  
}
```

In der `style.sass` müssen die Bibliotheken von Tailwind integriert werden:

```
@import 'tailwindcss/base'  
@import 'tailwindcss/components'  
@import 'tailwindcss/utilities'
```

Projekt starten

```
npm i  
npm start  
ng serve
```

Option	Beschreibung	Default-Wert
port	Port der Anwendung	4200
host	Host der Anwendung	localhost
watch	Live-Reload	true

```
ng serve --port=8080 --watch=false  
npm start -- --port=8080 --watch=false
```