

# Angular in 60 Minuten

Ein praxisnaher Einstieg für Java-Entwickler

**Demos @ Github**

<https://github.com/GEDOPLAN/EKJ-Angular60>

## Ich stelle mich vor

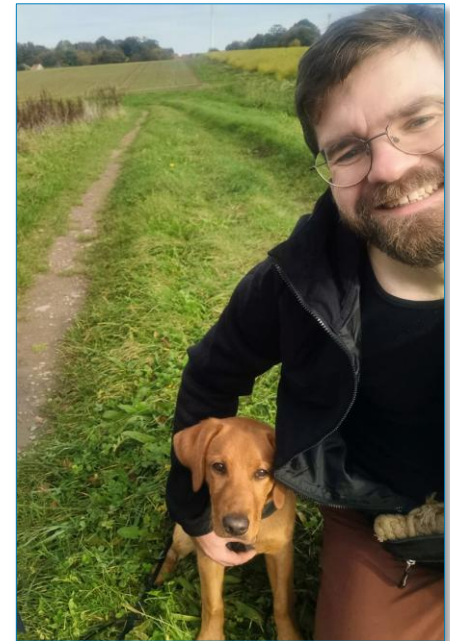
- Dominik Mathmann
- 15 Jahre unterwegs als Berater und Softwareentwickler in Projekten für und mit Kunden
- Fullstack-Entwickler und „GUI-Hansel“
- Schwerpunkte: Angular, JEE und Spring Boot



<https://www.linkedin.com/in/dominik-mathmann-a6861b332/>



[dominik.mathmann@gedoplan.de](mailto:dominik.mathmann@gedoplan.de)



## Warum Angular?

- Frontend-Framework
  - Enterprise tauglich , Komplettlösung
    - *Routing, HTTP, Forms, DI, Build*
  - Klare Struktur / Architekturvorgaben
- Objektorientierte Konzepte mit Typescript
- Weit verbreitet in größeren Unternehmen



## Angular Historie

Version	Jahr	
1	2010	AngularJS
2	2016	Komponenten, TypeScript
...		
4	2017	AOT Compiler, HttpClient, Template-Optimierung
...		
9	2020	Ivy Renderer, Template Type Checking
...		
17	2023	Neue Control Flow Syntax, M3, Signals
...		
21	2025	Aktuelle Version

## Setup

- Node.js
  - „Node.js® is a free, open-source, cross-platform JavaScript runtime environment“
- npm
  - „...the center of JavaScript code sharing“
- Angular CLI
  - Command Line Interface für Angular Projekte
  - Projekt-Setup, Komponenten Generierung, produktiver Workflow

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



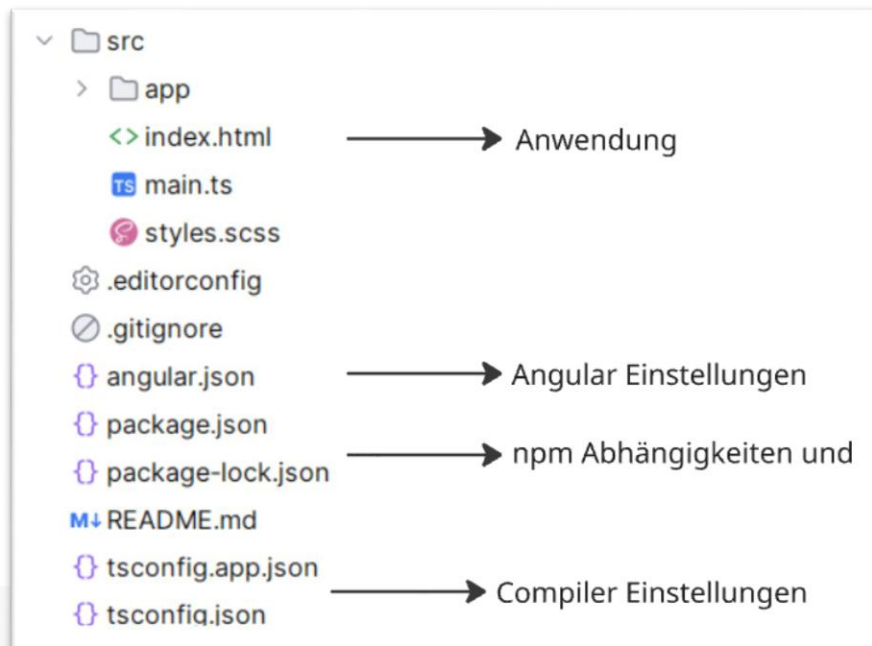
## Angular Projekt

- Ein neues Projekt anlegen

```
ng new
```

- Dev-Server starten

```
ng serve
```



## Bausteine

- Component
  - Template + Klasse + Style
- Service
  - Logik, Datenzugriffe
- Pipe
  - Formatierte Ausgaben
- Direktive
  - Verhalten/Optik von Komponenten anpassen
- Environment, Guard, Resolver, Router, Interceptor



# Component

- Anlage

ng g c [NAME]

user-list-filter

- user-list-filter.html
- user-list-filter.scss
- user-list-filter.ts

```
user-list-filter.html x
<h2>Filter</h2>
<input [placeholder]="placeholderText" [(ngModel)]="filterValue">
<button (click)="executeFilter()">{{ buttonText }}</button>
```

```
user-list-filter.scss x
h2{
  font-size: 12px;
}
```

```
user-list-filter.ts x
import ...
@Component({ Show usages & Dominik Mathmann *
  selector: 'app-user-list-filter',
  imports: [
    FormsModule
  ],
  templateUrl: './user-list-filter.html',
  styleUrls: ['./user-list-filter.scss'],
})
export class UserListFilter {...}
```

```
user-list.html x
<app-user-list-filter>
```

## Template Syntax

- Interpolation
  - `{{variable}}`
- Property Binding (*@Input*)
  - `[property] = „value“`
- Event Binding (*@Output*)
  - `(event) = „method()“`
- Two-Way Binding
  - `[(ngModel)] = „variable“`



## Komponenten - Datenfluss

```
export class UserListFilter {  
  
  @Input()  
  buttonText :string = 'filter';  
  @Input()  
  placeholderText :string = '';  
  
  @Output()  
  filter :EventEmitter<string> = new EventEmitter<string>();  
}
```

```
<app-user-list-filter placeholderText="Super Filter" (filter)="filter($event)"></app-user-list-filter>
```



# Service

- Anlage

ng g s [NAME]

```
user-data.ts x
> import ...

@Injectable({ Show usages  Dominik Mathmann
  providedIn: 'root',
})
export class UserData {

  http :HttpClient = inject(HttpClient);


  loadUsers(firstName?: string) :Observable<any> { Show usages  Dominik Mathmann
    return this.http.get<any>('https://dummyjson.com/users/search?q=' + firstName);
  }
}
```

```
user-list.ts x
> import ...

@Component({selector: 'app-user-list'...})
export class UserList {

  userDataService :UserData = inject(UserData); ←
  userListResponse :Observable<any> = this.userDataService.loadUsers( firstName: '' );
}
```

## Service

- Implementiert Logik und Datenzugriffe
- Verwendung via Injection ( *inject*(UserData) )
- In aller Regel global Bereitgestellt, Singleton ( **providedIn: 'root'** )
- Datenzugriff-Services Generierung via **OpenAPI Specification** 

## Was gibt's noch?

- Routing
- Angular Material Components
- Testing mit Vitest
- Pipes und Directives
- Content-Projektion
- Interceptoren
- Lifecycle Methoden
- Signals- und Reactive-Forms
- Lazy Loading von Komponenten
- I18n
- ...

