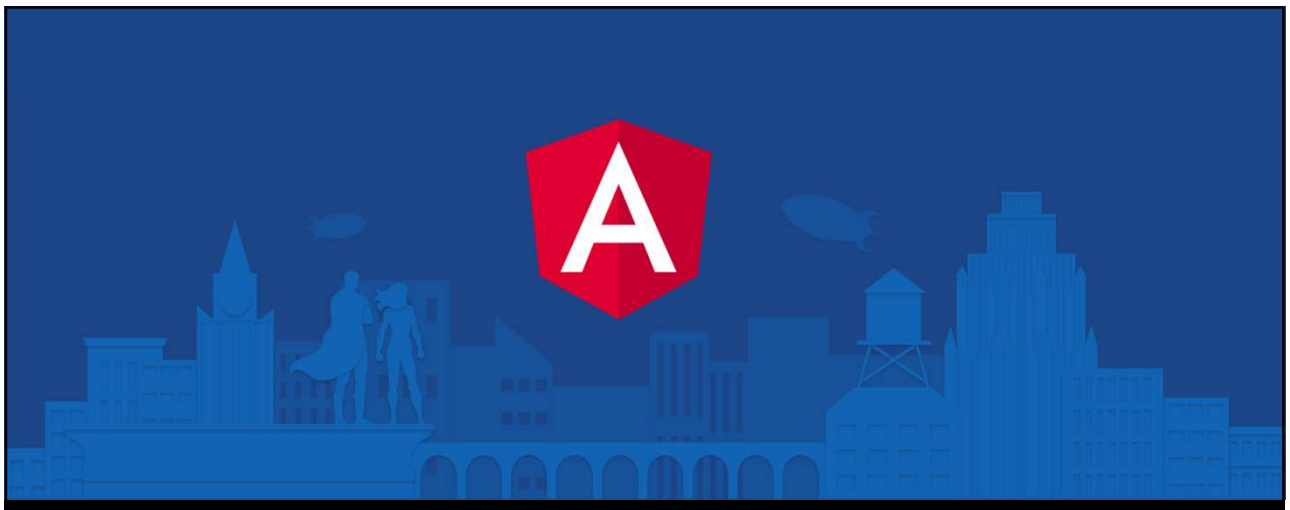


Angular: Erste Komponente

Michael Zikes
SOFTWAREarchitekt.at

Inhalt

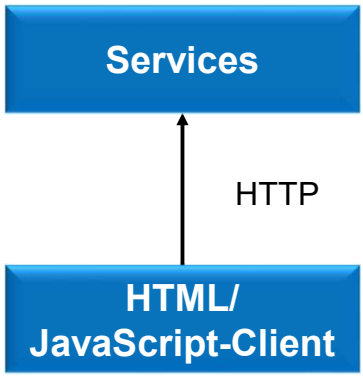
- Motivation
- Erste Schritte
- Eine erste Komponente
- HTTP-Zugriff

Motivation

Plattformen und Usability

HTML + JavaScript

Single Page Application (SPA)



SOFTWARE architekt.at

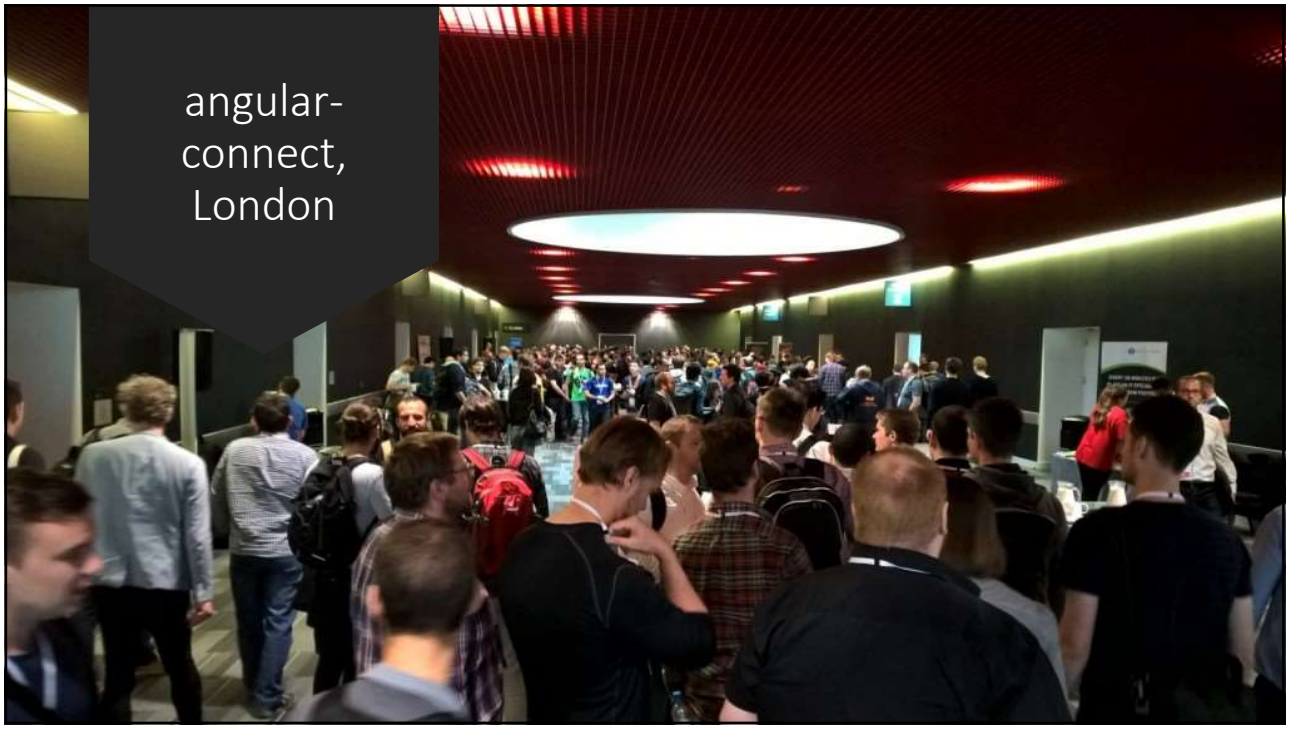
Frameworks machen SPA beherrschbar

Google

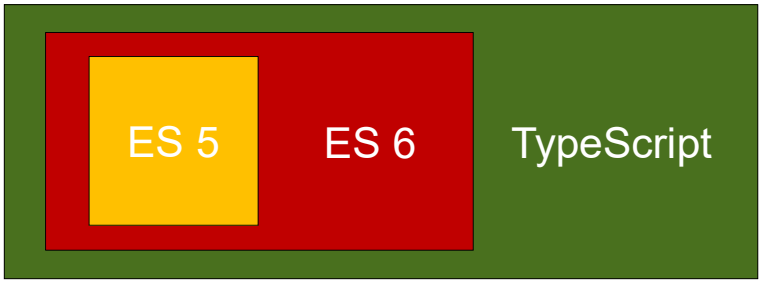
Community

Angular (2/4/5): >1,2M Devs


angular-
connect,
London
RANGLE.IO
ANGULAR
CONNECT


angular-
connect,
London

Sprachen



Kompilierung

SOFTWARE architekt.at

Erste Schritte mit Angular

AppComponent

```
@Component({
    selector: 'flug-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hallo Welt!';
}
```


AppComponent

```
import { Component } from '@angular/core';

@Component({
    selector: 'flug-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hallo Welt!';
}
```

Bibliothek
Beispiel: @angular/core

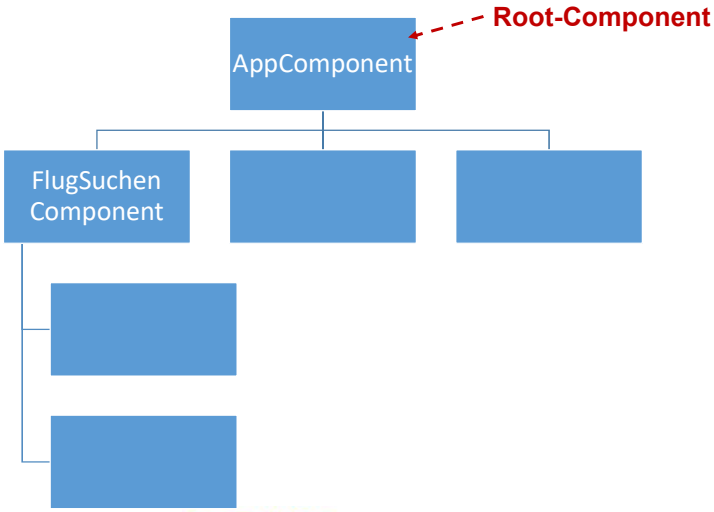
Eigenes Projekt
Beispiel: ../entities/flug
Keine Endung .ts

AppComponent

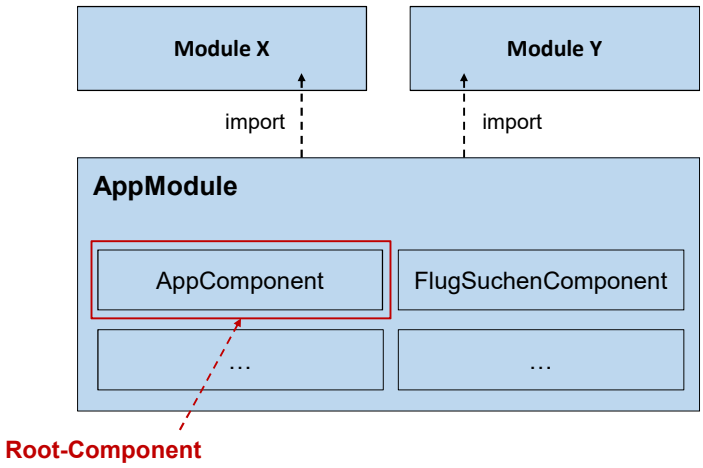
```
import { Component } from '@angular/core';

@Component({
    selector: 'flug-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hallo Welt!';
}
```

```
<h1>{{title}}</h1>
<div class="container">
    <flug-suchen></flug-suchen>
</div>
```


Anwendung == Kompontentenbaum



Module



AppModule

```
@NgModule({
    imports: [
       BrowserModule, HttpClientModule, FormsModule
    ],
    declarations: [
       AppComponent, FlugSuchenComponent
    ],
    bootstrap: [
       AppComponent
    ]
})
export class AppModule {
}
```

Bootstrapping

- Angular starten
- RootModule mit RootComponent bekannt geben

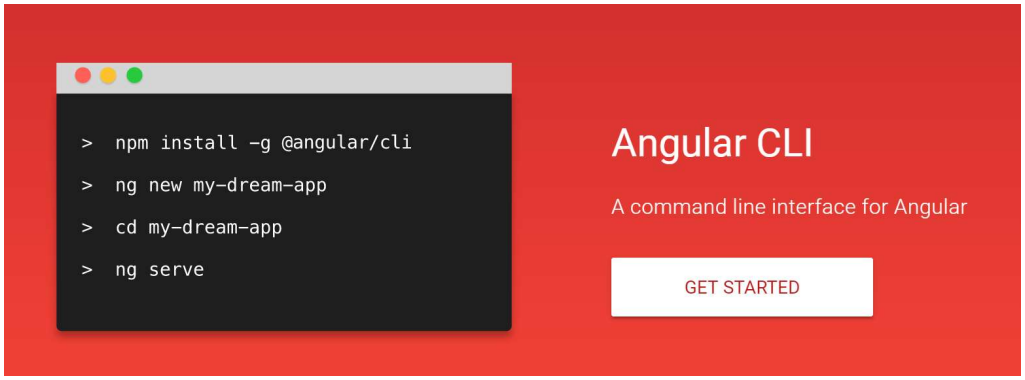
Bootstrapping

```
platformBrowserDynamic().bootstrapModule(AppModule);
```

index.html

```
[…]
<body>
  <flug-app></flug-app>
  <script src="..."></script>
</body>
[…]
```


Projektstart



Angular CLI

Unser Starterkit

- ng new starter
- cd starter
- npm i bootstrap --save
- Globale Styles in *angular.json* eintragen

```
[...]
"styles": [
    "styles.css",
    "../node_modules/bootstrap/dist/css/bootstrap.css",
   [...]
],
[...]
```

DEMO

SOFTWARE architekt.at

Eine erste Komponente

Komponente als TypeScript-Klasse

```
@Component({
    selector: 'flug-suchen',
    templateUrl: './flug-suchen.html'
})
export class FlugSuchenComponent {

    von: string;
    nach: string;
    fluege: Array<Flug>;

    search(): void { [...] }
    select(flug: Flug): void { [...] }
}
```


Template

Two-Way-Binding

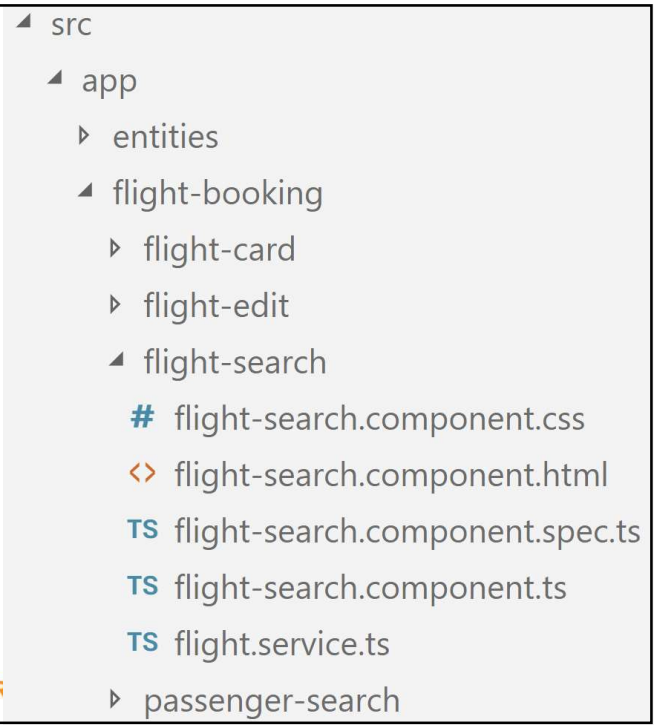
```
<input [(ngModel)]="von">
<input [(ngModel)]="nach">

<button [disabled]="!von || !nach" (click)="search()">
    Search
</button>

<table>
    <tr *ngFor="let flug of fluege">
        <td>{{flug.id}}</td>
        <td>{{flug.datum}}</td>
        <td>{{flug.von}}</td>
        <td>{{flug.nach}}</td>
    </tr>
</table>
```

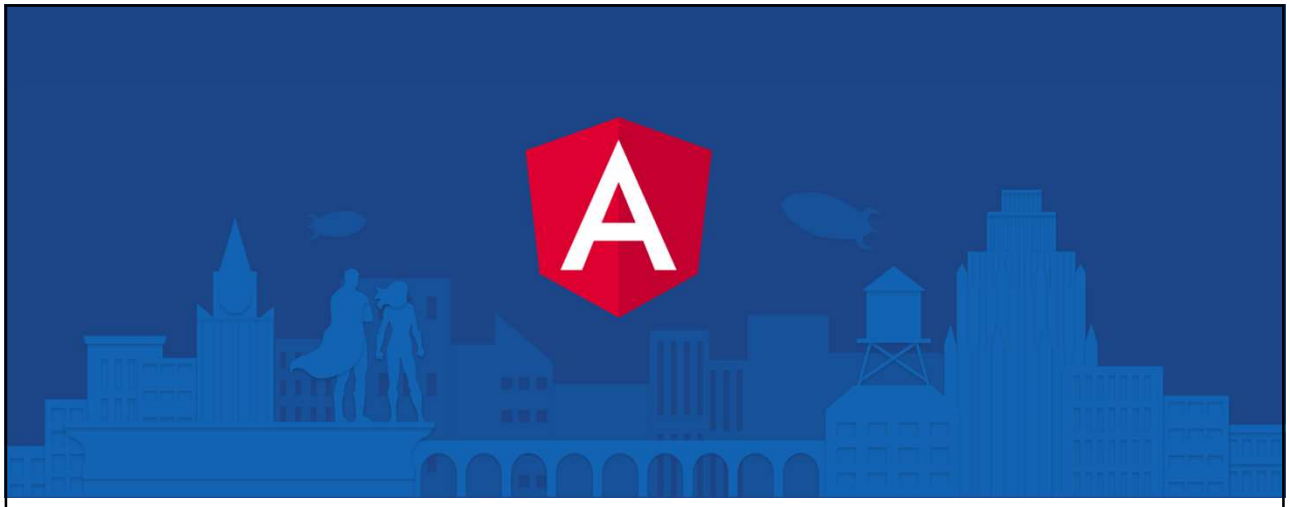
Event-Binding

Property-Binding

Template

DEMO





Auf HTTP-Ressourcen zugreifen

HttpClient

- get(url, options)
- post (url, body, options)
- put(url, body, options)
- delete(url, options)
- …

HttpClient

- get<T>(url, options)
- post<T>(url, body, options)
- put<T>(url, body, options)
- delete<T>(url, options)
- …

HttpClient injizieren

```
@Component({
    selector: 'flug-suchen',
    templateUrl: './flug-suchen.html'
})
export class FlugSuchenComponent {

    von: string;
    nach: string;
    fluege: Array<Flug>;

    constructor(http: HttpClient) { […] }

    search(): void { [...] }
    select(flug: Flug): void { [...] }
}
```


HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';
```

HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);
```


HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');
```

HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');

this.http
    .get<Flight[]>(url, { params: params, headers: headers })
```


HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');


this.http
    .get<Flight[]>(url, { params, headers })
```

HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');


this.http
    .get<Flight[]>(url, { params, headers })
    .subscribe(
        function(flights) { […] }
    );
```


HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');

let that = this;
this.http
    .get<Flight[]>(url, { params, headers })
    .subscribe(
        function(flights) {
              that.flights = flights;
        }
);
```

HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');


this.http
    .get<Flight[]>(url, { params, headers })
    .subscribe(
        flights => {
              this.flights = flights;
        }
);
```


HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');


this.http
    .get<Flight[]>(url, { params, headers })
    .subscribe(
        flights => { this.flights = flights; },
        err => { console.error('Fehler beim Laden', err); }
    );
```

DEMO

HttpClient nutzen

```
let url = 'http://www.angular.at/api/flight';

let params = new HttpParams()
                      .set('from', this.from)
                      .set('to', this.to);

let headers = new HttpHeaders()
                      .set('Accept', 'application/json');

this.http
    .get<Flight[]>(url, { params, headers })
    .subscribe(
        flights => { this.flights = flights; },
        err => { console.error('Fehler beim Laden', err); }
    );
```

<- Observable

SOFTWARE architekt.at

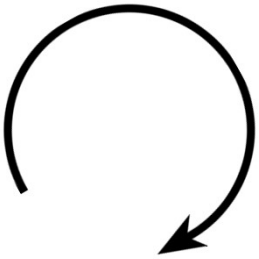
Observable
„Quelle“

Operator
(z. B. map)

Observer
„Senke“

SOFTWARE architekt.at

Observable



Observable

```
.subscribe(
   (result) => { … },
   (error) => { … },
   () => { … }
);
```

Observer

SOFTWARE architekt.at

DEMO