Angular: First Steps

Manfred Steyer
SOFTWAREarchitekt.at

1

Agenda

- Motivation
- First steps
- Our first Components
- HTTP access

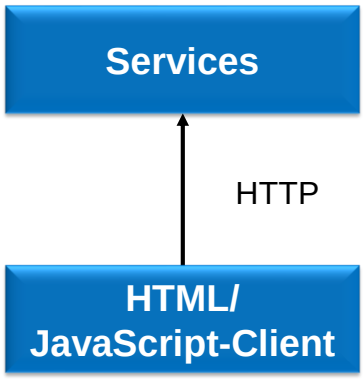
3

Motivation

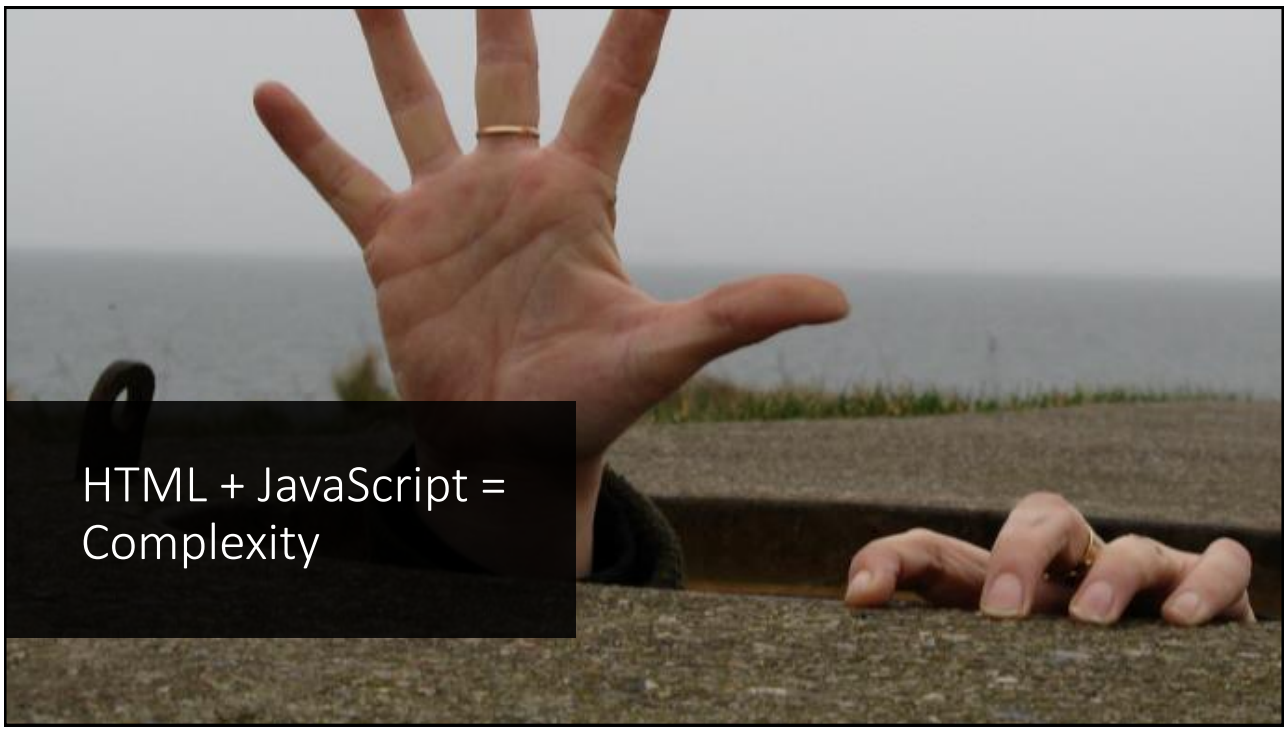
Plattformen und Usability

HTML + JavaScript

Single Page Application (SPA)



6



7

Use of frameworks lead to manageable SPAs

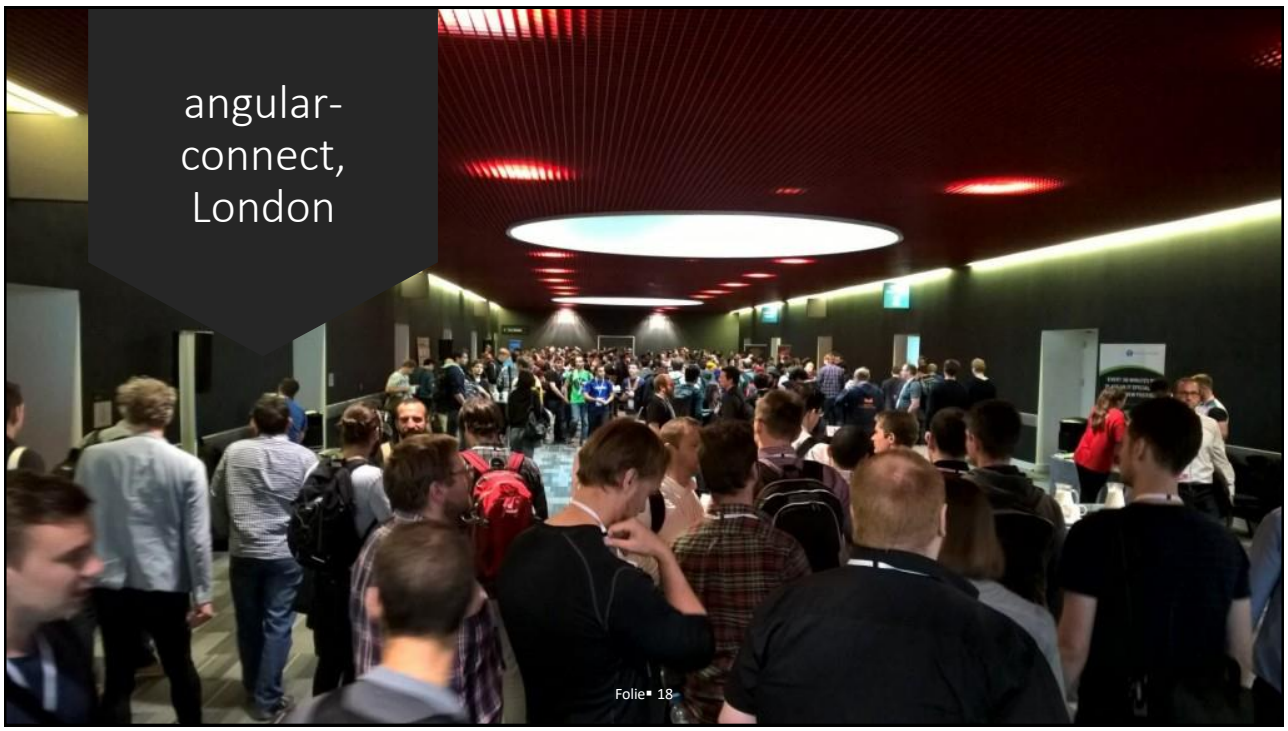
8

Google	Community
> 1500 Angular Projects at Google	Several Million Angular Devs wordwide

12

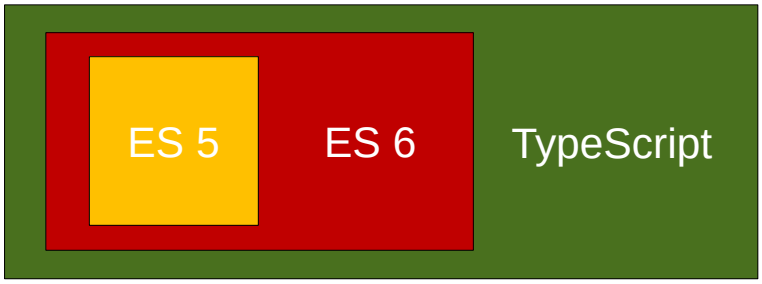

angular-
connect,
London
RANGLE.IO
ANGULAR
CONNECT
Folie▪ 17

17


angular-
connect,
London
Folie▪ 18

18

Programming languages



Compiling

23

First steps with Angular

32

AppComponent

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


AppComponent

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

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

Library
Example: @angular/core

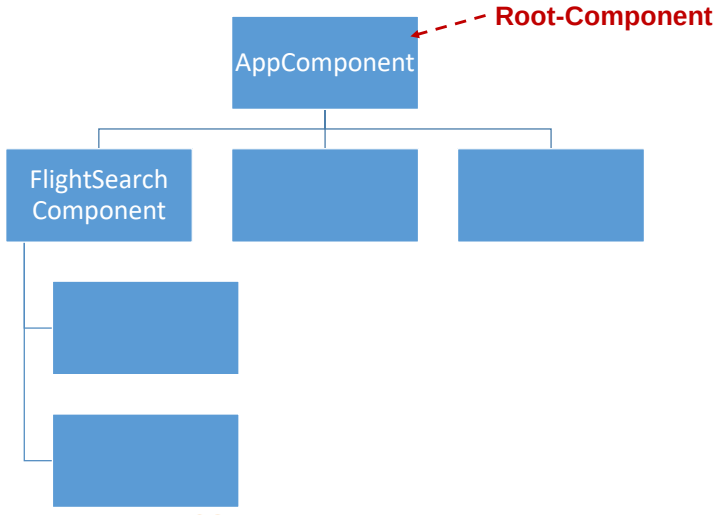
Custom project
Example: ../entities/flight
No file extension .ts

AppComponent

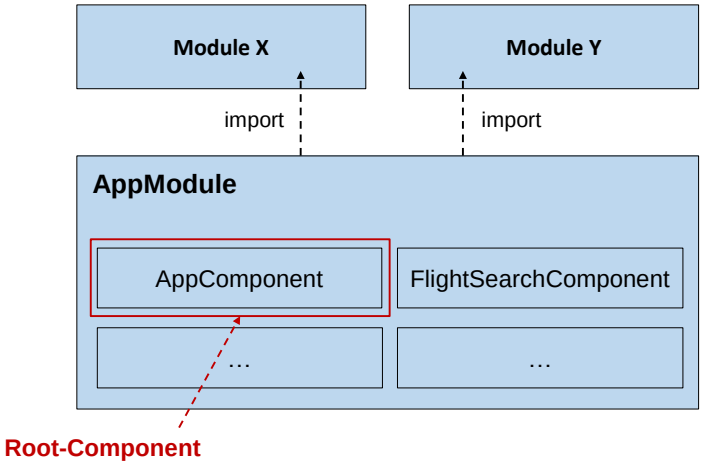
```
import { Component } from '@angular/core';

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

```
<h1>{{title}}</h1>
<div class="container">
    <flight-search></flight-search>
</div>
```


Application == Component Tree



Module



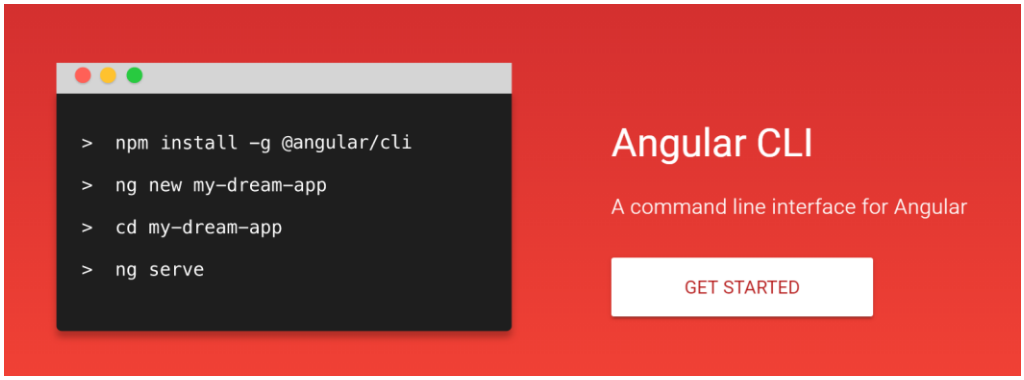
AppModule

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

index.html

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


Project Start



Angular CLI

57

How this starter kit was created

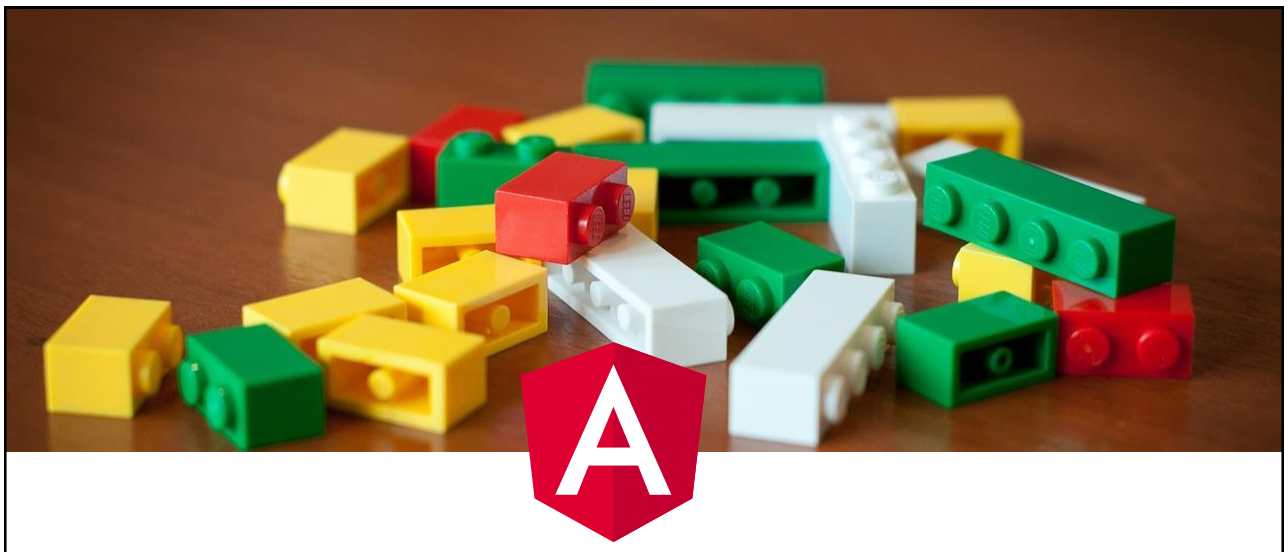
- ng new starter
- cd starter
- npm i bootstrap --save
- Add Files for Bootstrap Theme to Folder *assets*
- Insert global styles in .angular-cli.json

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


58

DEMO

61



Our first component

63

Component as TypeScript Class

```
@Component({
    selector: 'flight-search',
    templateUrl: './flight-search.html'
})
export class FlightSearchComponent {

    from: string;
    to: string;
    flights: Array<Flight>;

    search(): void { [...] }
    select(flight: Flight): void { [...] }
}
```


Template

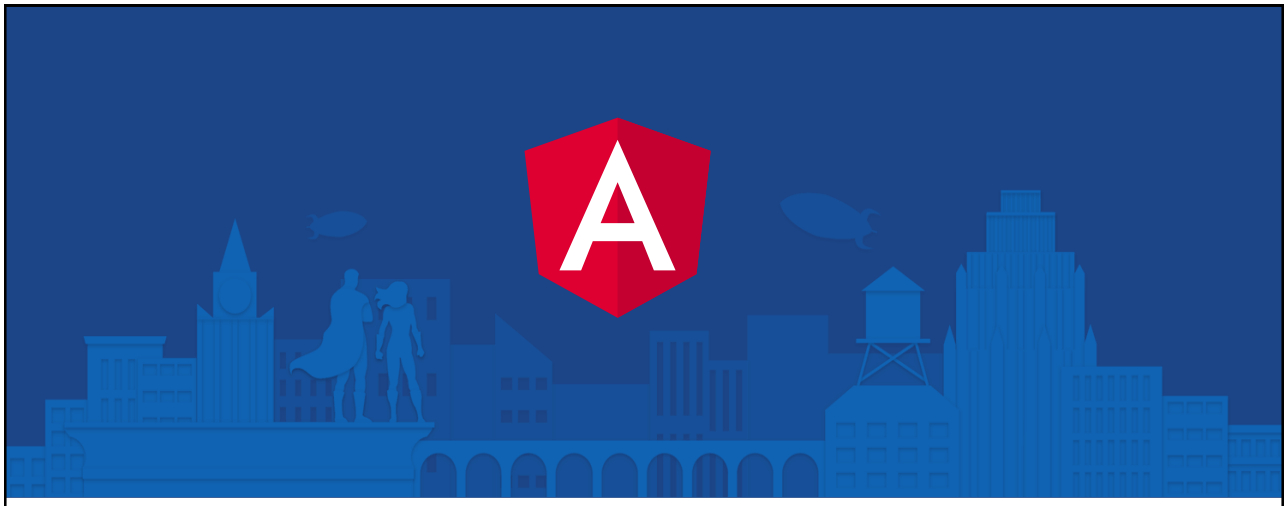
Two-Way-Binding

```
<input [(ngModel)]="from">
<input [(ngModel)]="to">

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

<table>
    <tr *ngFor="let flight of flights">
        <td>{{flight.id}}</td>
        <td>{{flight.date}}</td>
        <td>{{flight.from}}</td>
        <td>{{flight.to}}</td>
    </tr>
</table>
```

Event-Binding

Property-Binding

Template



Access HTTP resources

73

HttpClient

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

74

HttpClient

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

Inject HttpClient

```
@Component({
    selector: 'flight-search',
    templateUrl: './flight-search.html'
})
export class FlightComponent {

    from: string;
    to: string;
    flight: Array<Flight>;

    constructor(http: HttpClient) { […] }

    search(): void { [...] }
    select(flight: Flight): void { [...] }
}
```

Use HttpClient

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


79

Use HttpClient

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

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


80

Use HttpClient

```
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');
```


Use HttpClient

```
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 })
```

Use HttpClient

```
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) { … }
);
```


84

Use HttpClient

```
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;
        }
);
```


85

Use HttpClient

```
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;
        }
);
```


86

Use HttpClient

```
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('Error on loading flights', err); }
    );
```


87

DEMO

98

Use HttpClient

```
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('Error on loading flights', err); }
    );
```

<- Observable

99

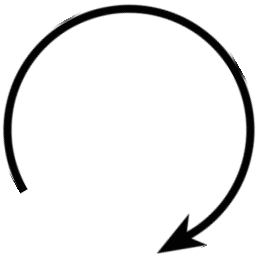
Observable
„source“

Operator
(e.g. map)

Observer
„trough“

101

Observable



Observable

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

Observer

102

Map operator

```
this
    .http
    .get(…)
    .map(flights => mapToOffers(flights))
    .subscribe(
        (offers) => { … },
        (err) => { console.error(err); }
    );
```


DEMO