ManfredSteyer

Angular: Erste Schritte

Manfred Steyer
SOFTWAREarchitekt.at

Inhalt

SOFTWAREarchitekt.at

Motivation

Page ▪ 3

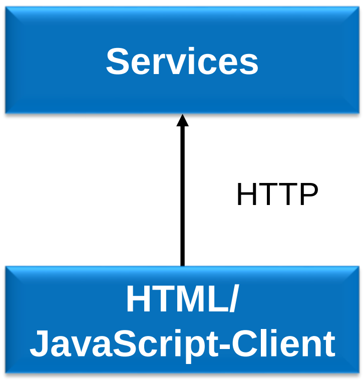
SOFTWARE architekt.at

Plattformen und Usability

HTML + JavaScript

SOFTWARE architekt.at

Single Page Application (SPA)



SOFTWARE architekt.at

HTML + JavaScript =
Komplexität

Frameworks machen SPA beherrschbar

Page ▪ 7

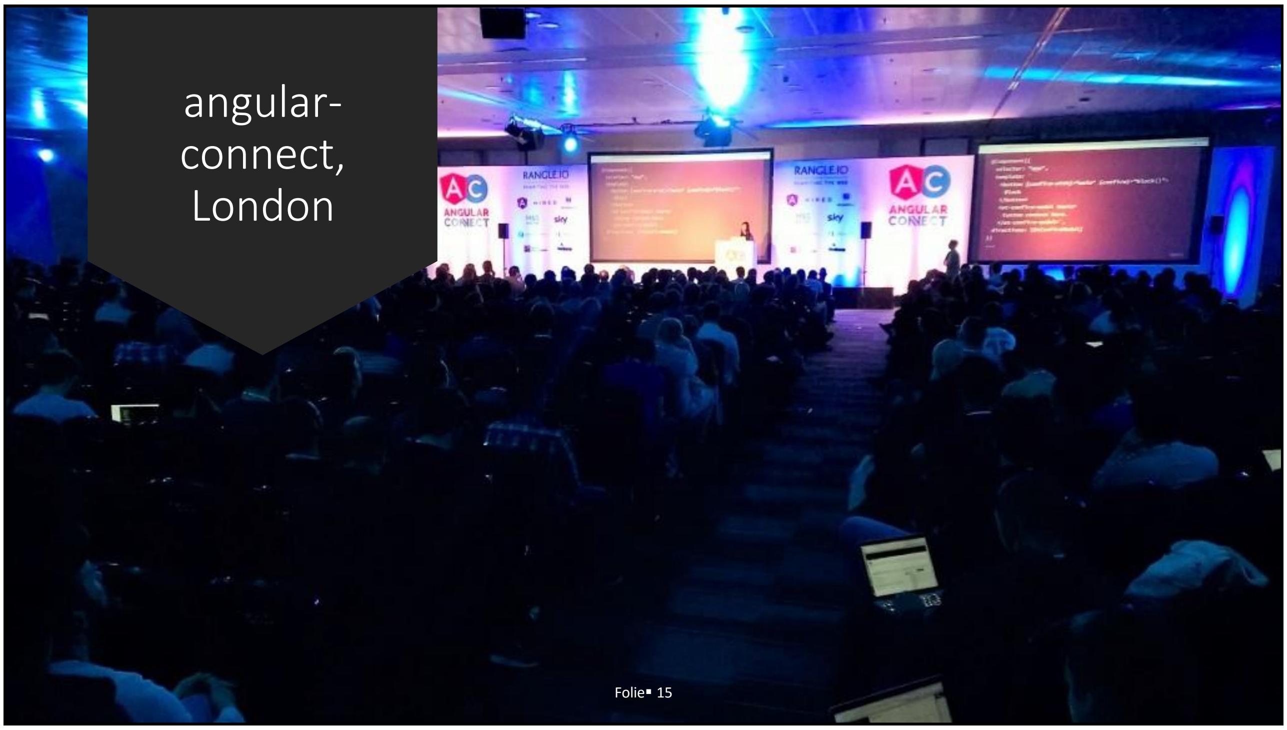
Google

Community

v1: 1,2 Mio Entwickler Weltweit

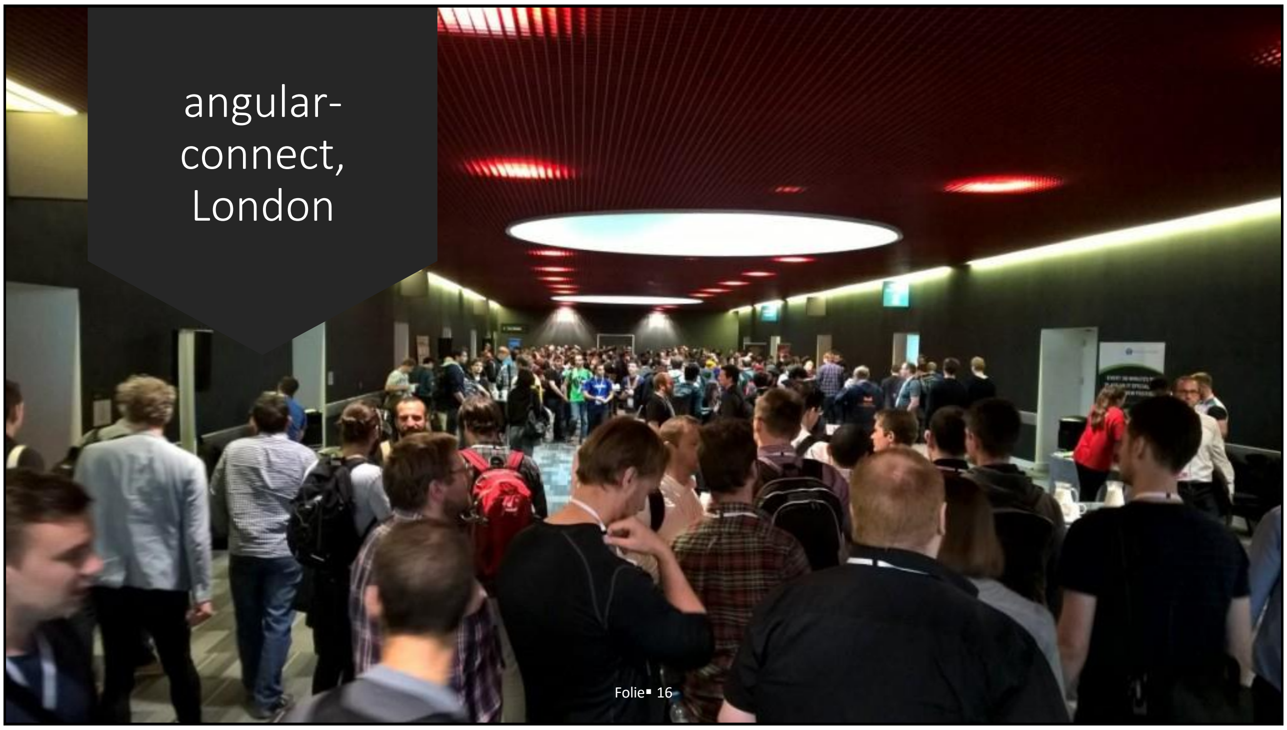
v2: 600K Entwickler

angular-connect, London



Folie 15

angular-connect, London

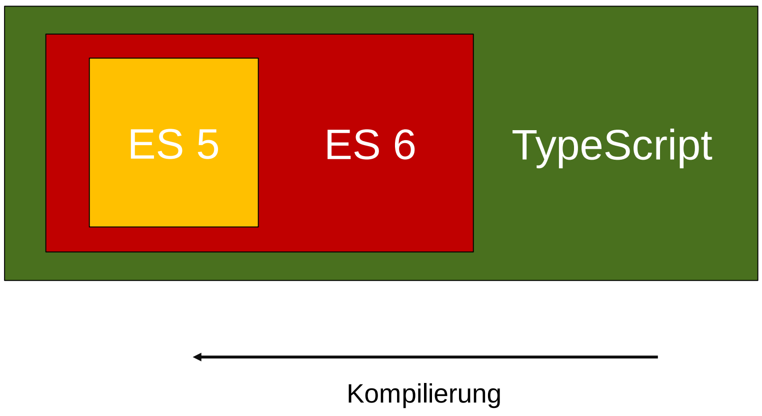


Folie 16

Was ist Angular?

- SPA-Framework
- Performance
- Komponenten
- Modern Web
- Flexibler Renderer

Sprachen



Eine erste Komponente

Page ▪ 30

Komponente als TypeScript-Klasse

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

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

    constructor(http: Http) { }

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

Page ▪ 32

SOFTWARE architekt.at

Andere Dateien (Module) referenzieren

```
import { Component } from '@angular/core';
import { Flug } from '../entities/flug';

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

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

    […]
}
```

Bibliothek
Beispiel: @angular/core

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

SOFTWARE *architekt.at*

Datentypen

number | string | boolean | Function
Object | any | null | undefined
Klassen | Interfaces

Page ▪ 34

SOFTWARE *architekt.at*

Interface Flug

```
export interface Flight {
    id: number;
    from: string;
    to: string;
    date: string;
}
```

```
search(): void {
    let f1: Flight = { id: 4711, from: 'Graz', to: 'Hamburg', date: '...' };
    […]
}
```

Strukturelle Zuweisungskompatiblität
("Duck Typing")

SOFTWARE architekt.at

Access-Modifier (TypeScript-Erweiterung)

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

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

    constructor(http: Http) { }

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

SOFTWARE architekt.at

Access Modifier (TypeScript-Erweiterung)

public | protected | private | readonly

Nur eigene Klasse und Subklassen

Komponente als TypeScript-Klasse

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

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

    constructor(http: Http) { }

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

Template

Two-Way-Binding

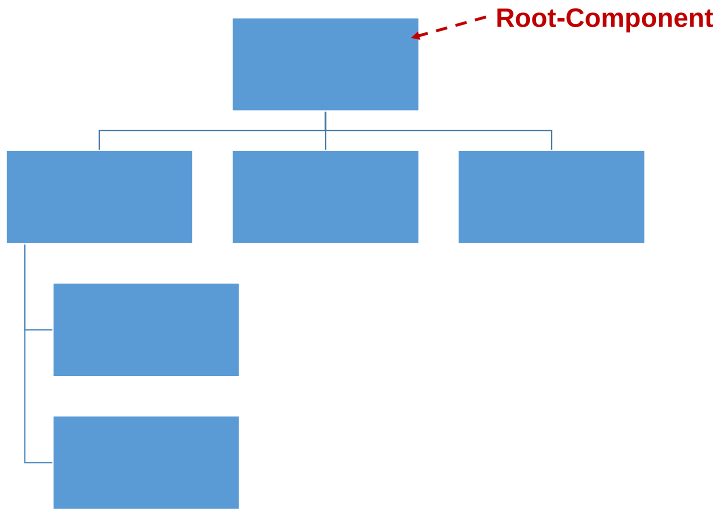
```
<input [(ngModel)]="von">
<input [(ngModel)]="nach">

<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

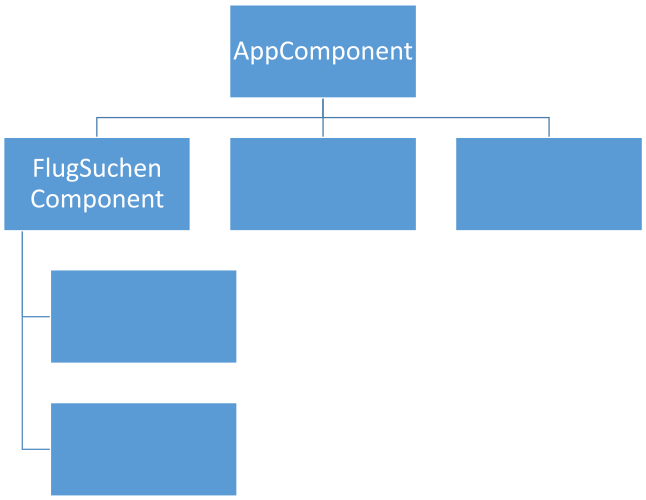
Property-Binding

Template

Anwendung == Kompontentenbaum



Anwendung == Kompontentenbaum



SOFTWARE architekt.at

AppComponent

```
@Component({
    selector: 'flug-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
}
```

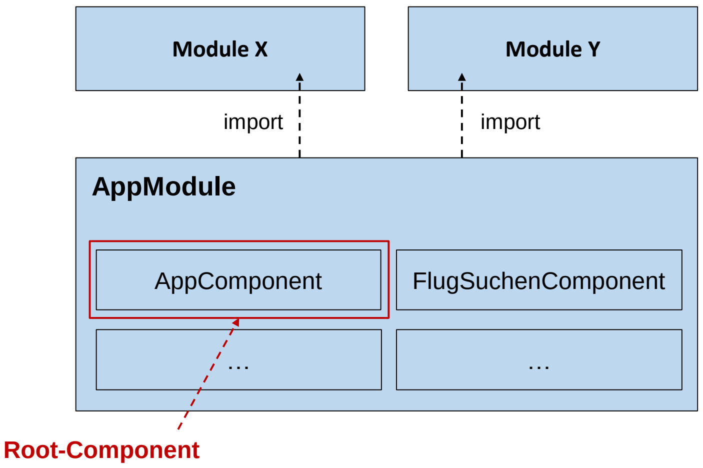
SOFTWARE architekt.at

AppComponent

```
<nav class="...">
    <!-- Menü -->
</nav>

<div class="container">
    <flug-suchen></flug-suchen>
</div>
```

SOFTWARE architekt.at

Module



SOFTWARE architekt.at

AppModule

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

Bootstrapping

- Angular starten
- RootModule mit RootComponent bekannt geben

Bootstrapping

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

SOFTWARE architekt.at

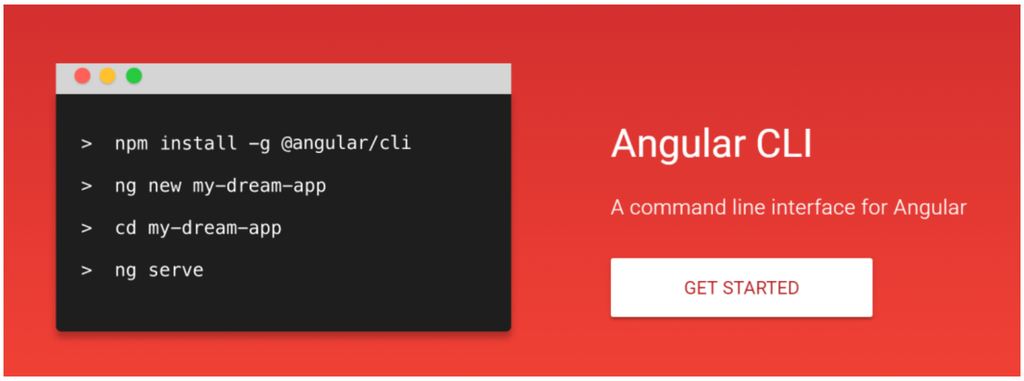
index.html

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

SOFTWARE architekt.at

Projektstart

SOFTWARE architekt.at



```
> npm install -g @angular/cli
> ng new my-dream-app
> cd my-dream-app
> ng serve
```

Angular CLI

A command line interface for Angular

Angular CLI

Starterkit für diesen Workshop

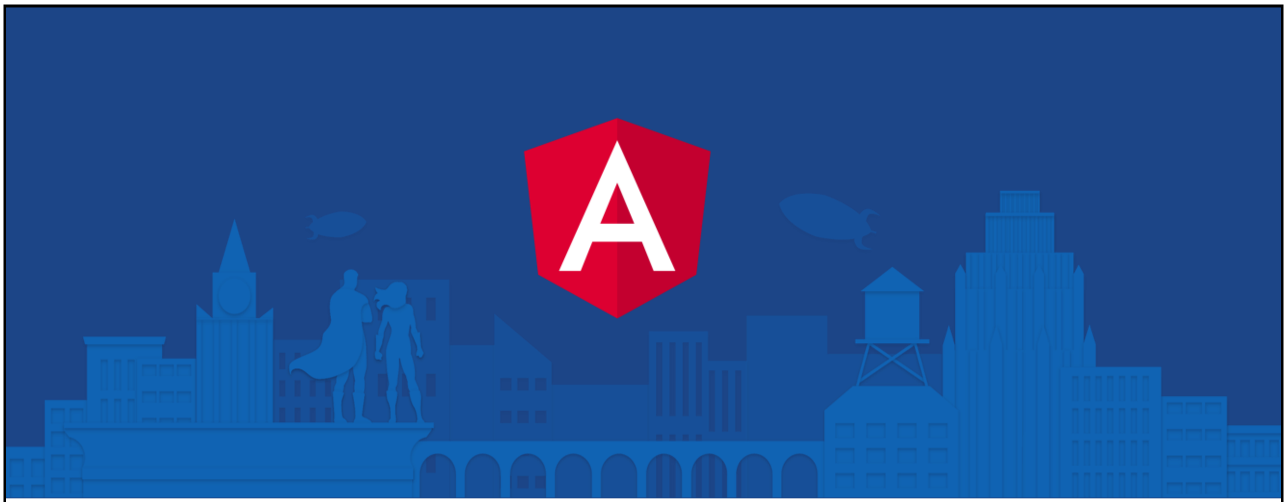
- Generiert mit CLI
- Bootstrap für Styling (CSS)
  - npm i bootstrap --save
- Ein paar eigene Styles
  - styles.css

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

SOFTWARE architekt.at

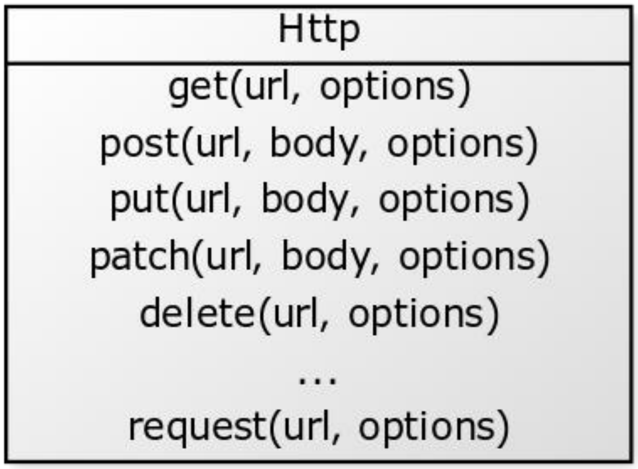
DEMO

SOFTWARE architekt.at



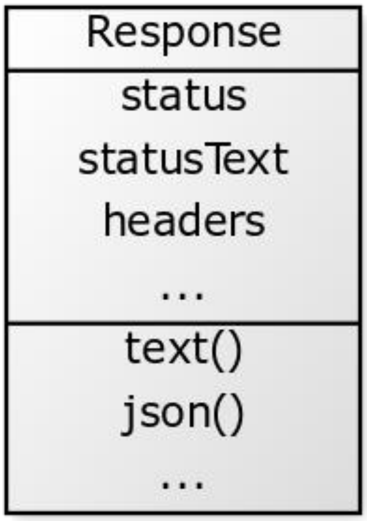
Auf HTTP-Ressourcen zugreifen

Http-Service



```
Http
get(url, options)
post(url, body, options)
put(url, body, options)
patch(url, body, options)
delete(url, options)
...
request(url, options)
```

Response



Http-Service

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

SOFTWARE architekt.at

Http-Service

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

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

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);
```

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

this
    .http
    .get(url, { headers: headers, search: search })
```

SOFTWARE architekt.at

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

this
    .http
    .get(url, { headers, search })
```

SOFTWARE architekt.at

Http-Service

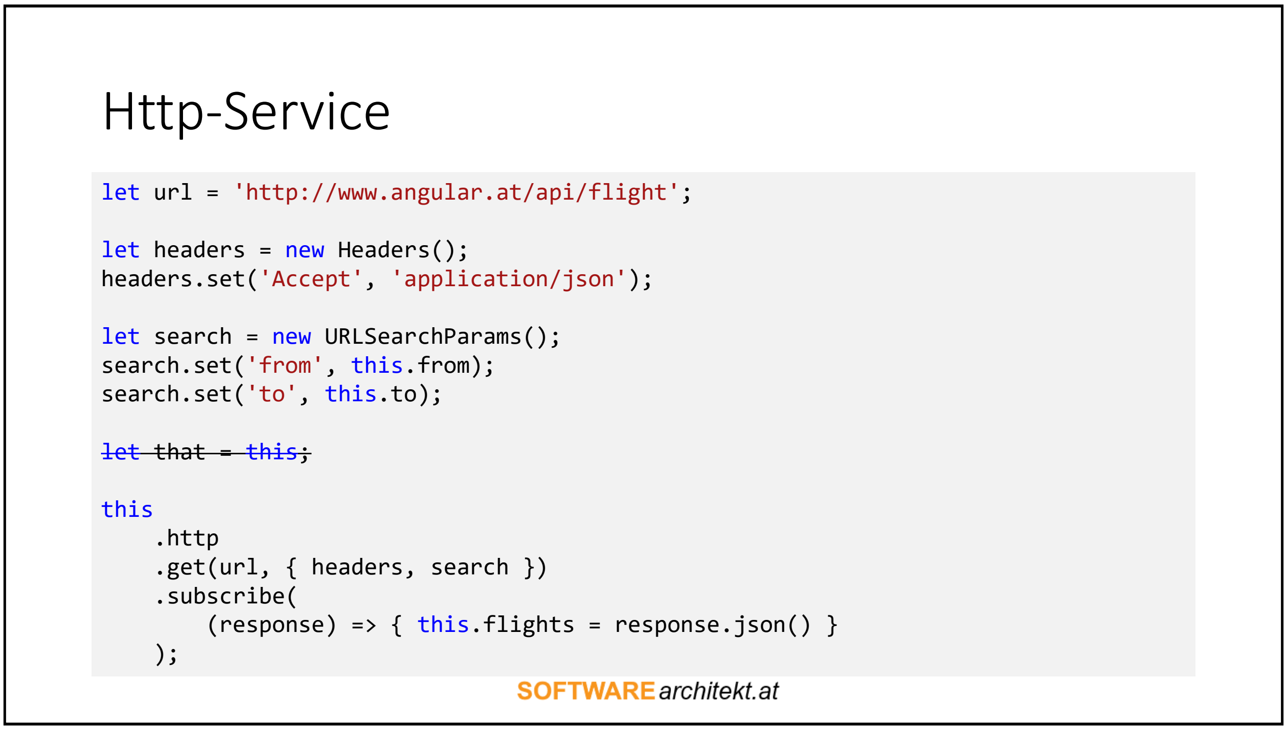
```
let url = 'http://www.angular.at/api/flight';

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

this
    .http
    .get(url, { headers, search })
    .subscribe(
        function (response) { … }
    );
```

SOFTWARE*architekt.at*

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

let that = this;

this
    .http
    .get(url, { headers, search })
    .subscribe(
        function (response) { that.flights = response.json() }
    );
```

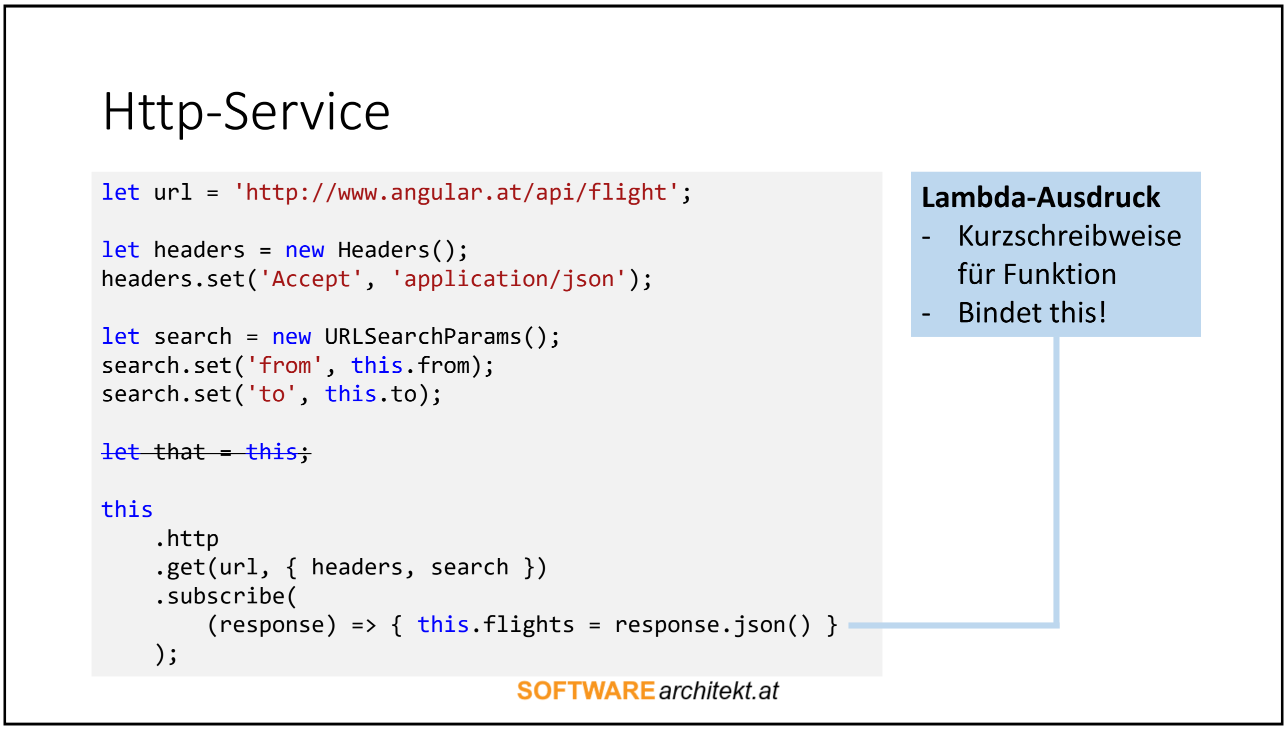
SOFTWARE*architekt.at*



Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

~~let that = this;~~

this
    .http
    .get(url, { headers, search })
    .subscribe(
        (response) => { this.flights = response.json() }
    );
```

SOFTWARE architekt.at



Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

~~let that = this;~~

this
    .http
    .get(url, { headers, search })
    .subscribe(
        (response) => { this.flights = response.json() }
    );
```

Lambda-Ausdruck
- Kurzschreibweise für Funktion
- Bindet this!

SOFTWARE architekt.at

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

this
    .http
    .get(url, { headers, search })
    .subscribe(
        (response) => { … },
        (errResponse) => { … }
    );
```

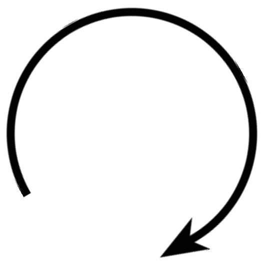
SOFTWARE*architekt.at*

Http-Service

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

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

let search = new URLSearchParams();
search.set('from', this.from);
search.set('to', this.to);

this
    .http
    .get(url, { headers, search })
    .subscribe(                          ┐
        (response) => { … },             │
        (errResponse) => { … }           │ <--------- Observable
    );                                   ┘
```

SOFTWARE*architekt.at*

Observable



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

Observable

Observer

Map-Operator

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

SOFTWARE architekt.at

DEMO

Page ▪ 83

SOFTWARE architekt.at