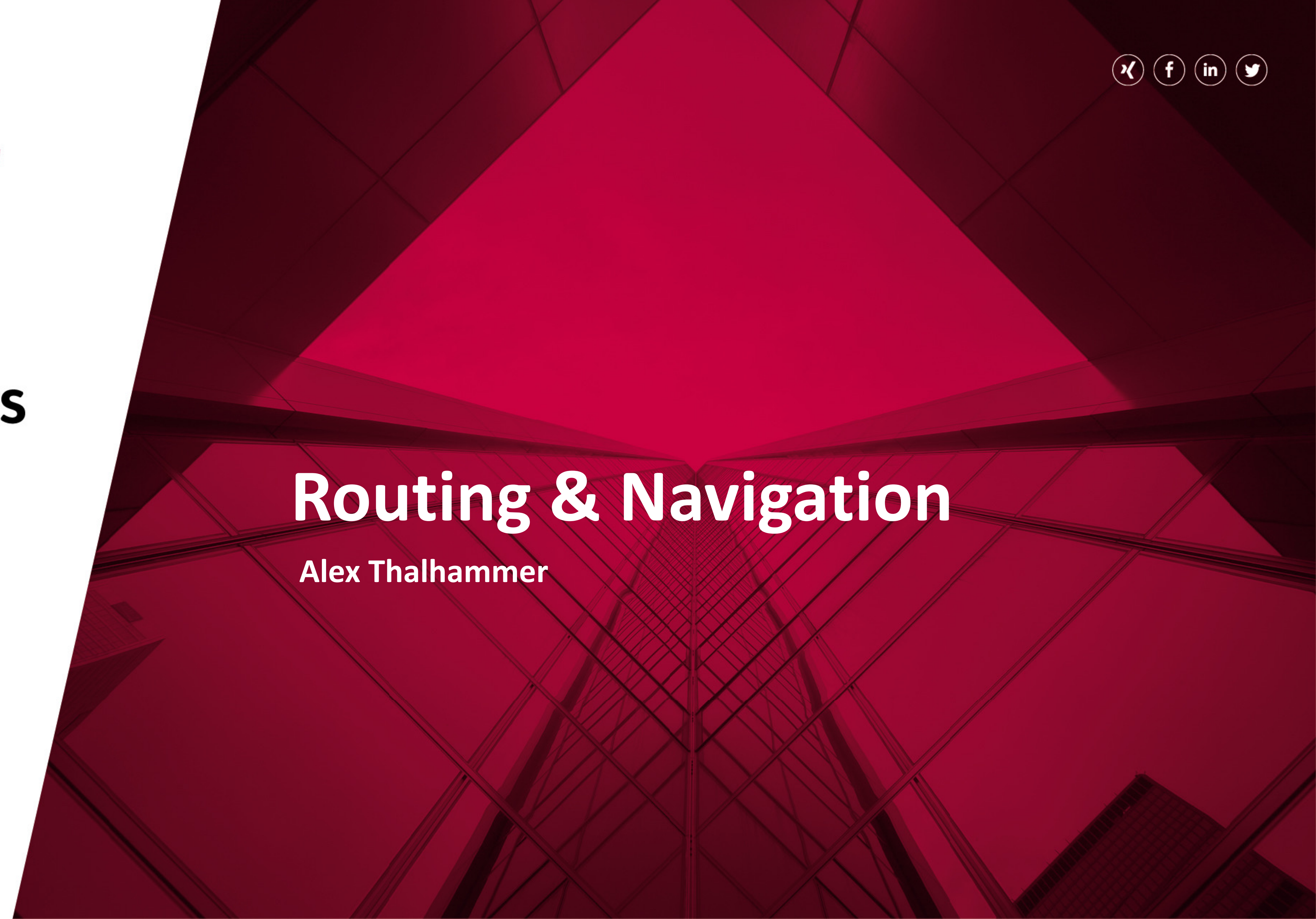
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



Routing & Navigation

Alex Thalhammer

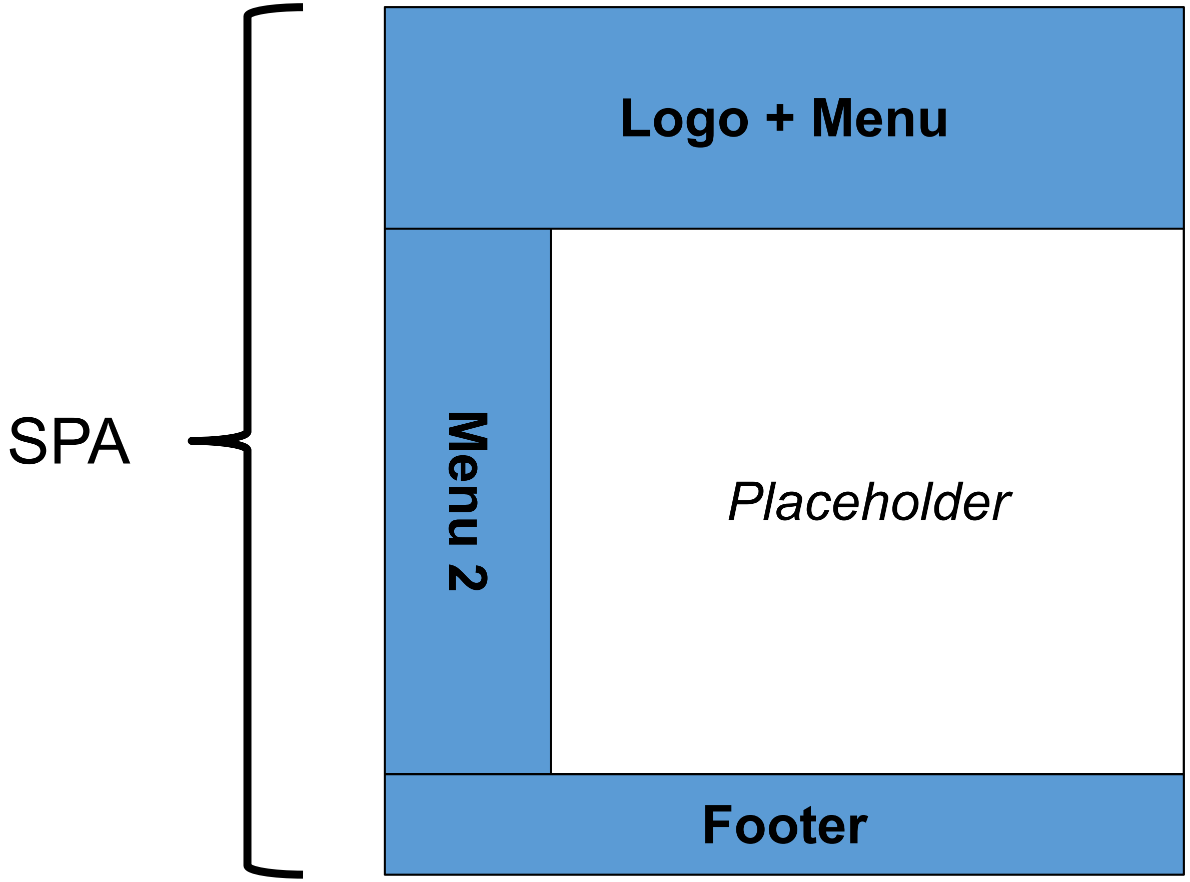
Outline

- Motivation
- Configuration
- Routing parameters

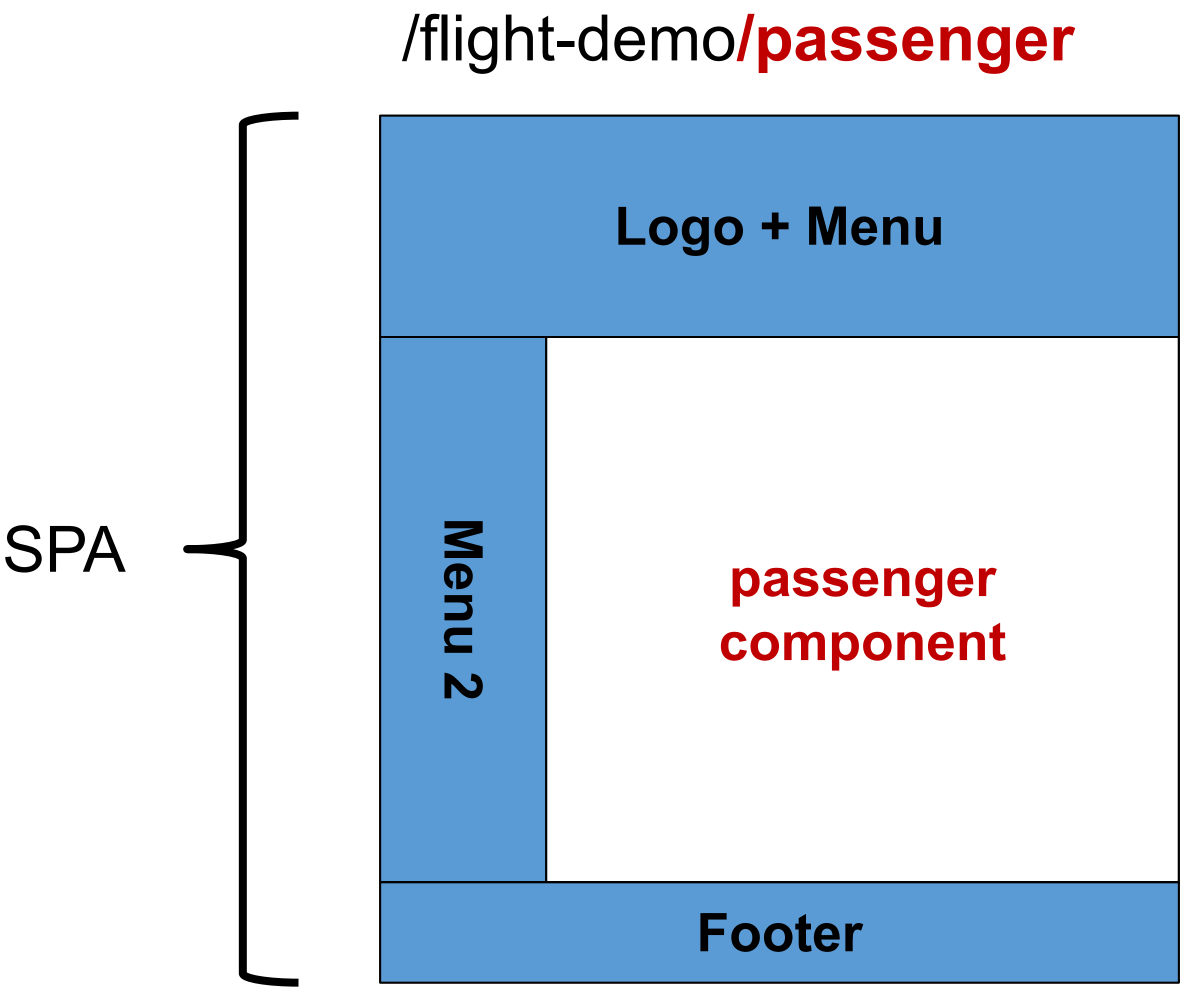
Motivation

- SPAs → single page application
- Simulate pages → routes
- URL should direct to the routed component
  - Bookmarks
  - **Back button**

Routing in Angular



Routing in Angular



Routing via hash fragment

- /flight-demo**#passenger**
- Hash-Fragment wird nie zum Server gesendet

Routen via History API

- /flight-demo/**passagier**
- Whole URL will be send to the server
- Server responds with the SPA
- Server can render the inital view
  - Performance, SEO, …
- SPA informs the browser which part of the URL belongs:
  - to the server
  - to the Angular Route

Angular Router

Preparation

- ES6/TypeScript-Module: @angular/router
- Angular Module: RouterModule
- Definition of the limit between URL and Angular Route with the History API:
    - Options to define:
        - angular.json
        - <base href="/"> in index.html
        - Token: APP_BASE_HREF (@angular/common)

Configuration

```
export const APP_ROUTES: Routes = [
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'flight-search',
        component: FlightSearchComponent
    }
]
```

Configuration

```
export const APP_ROUTES: Routes = [
    {
        path: '',
        redirectTo: 'home',
        pathMatch: 'full'
    },
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'flight-search',
        component: FlightSearchComponent
    }
]
```

Configuration

```
export const APP_ROUTES: Routes = [
    {
        path: '',
        redirectTo: 'home',
        pathMatch: 'full'
    },
    {
        path: 'home',
        component: HomeComponent
    },
    […]
    {
        path: '**',
        redirectTo: 'home'
    }
]
```

Last route!!!

Configuration

```
// app.module.ts
@NgModule({
    imports: [
        BrowserModule,
        HttpModule,
        FormsModule,
        RouterModule.forRoot(APP_ROUTES)
    ],
    […]
})
export class AppModule {}
```

for feature module: forChild

for root module

View von AppComponent

```
<a [routerLink]="'/'">Home</a>
<a [routerLink]="'/flight-search'">Search flight</a>

<div>
    <router-outlet></router-outlet>
</div>
```

View von AppComponent

```
<a routerLink="/">Home</a>
<a routerLink="/flight-search">Search flight</a>

<div>
    <router-outlet></router-outlet>
</div>
```

Programatic routing

```
export class AppComponent {

    constructor(private router: Router) {}

    navigateToFlightSearch(): void {
        this.router.navigate(['/flight-search']);
    }

}
```

DEMO

Routing with parameters

Parameters

- passenger/7
- passenger/7;details=true;page=7
- passenger/7;details=true;page=7/flights

Configuration

```
const APP_ROUTES: Routes = [

    […]

    {
        path: 'flight-search',
        component: FlightSearchComponent
    },
    {
        path: 'flight-edit/:id',
        component: FlightEditComponent
    }
}
```

Parse parameters

```
export class FlightEditComponent {
    public id: string;

    constructor(private route: ActivatedRoute) {

    ngOnInit(): void {
       this.route.paramMap.subscribe(
            params => {
                this.id = params.get('id');
                […]
            }
        );
    }
    […]
}
```

Links of routes with parameters

```
<a [routerLink]="['/flight-edit', flight.id]">Edit</a>
```

Links of routes with parameters

```
<a [routerLink]="['/flight-edit', flug.id, {showDetails:true}]">Edit</a>
```

DEMO

LAB

Lazy Loading

Why Lazy Loading?

- Load modules when they are needed
- Improve initial load (performance → very important!)

Root module with Lazy Loading

```
const APP_ROUTE_CONFIG: Routes = [
    {
        path: '',
        redirectTo: 'home',
        pathMatch: 'full'
    },
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'flights',
        loadChildren: () =>
             import('[…]flight-booking.module')
               .then(m => m.FlightBookingModule)

    }
];
```

Routes for feature module

```
const FLUG_ROUTES =    [
    {
        path: 'subroute',
        component: FlightBookingComponent,
        […]
    },
    […]
}
```

```
export const FlugRouterModule =
                   RouterModule.forChild(FLUG_ROUTES);
```

DEMO

Preloading

Idea

- Once the initial load (the important one) is complete load the lazy loaded modules (before they are even used)

- Once the module will come into use it's immediately accessable

Use preloading (very easy!)

```
export const AppRoutesModule =
                 RouterModule.forRoot(
                   ROUTE_CONFIG,
                   { preloadingStrategy: PreloadAllModules });
```

DEMO

LAB