



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Routing

Alex Thalhammer

# Outline

- Motivation
- Configuration
- Routing parameters

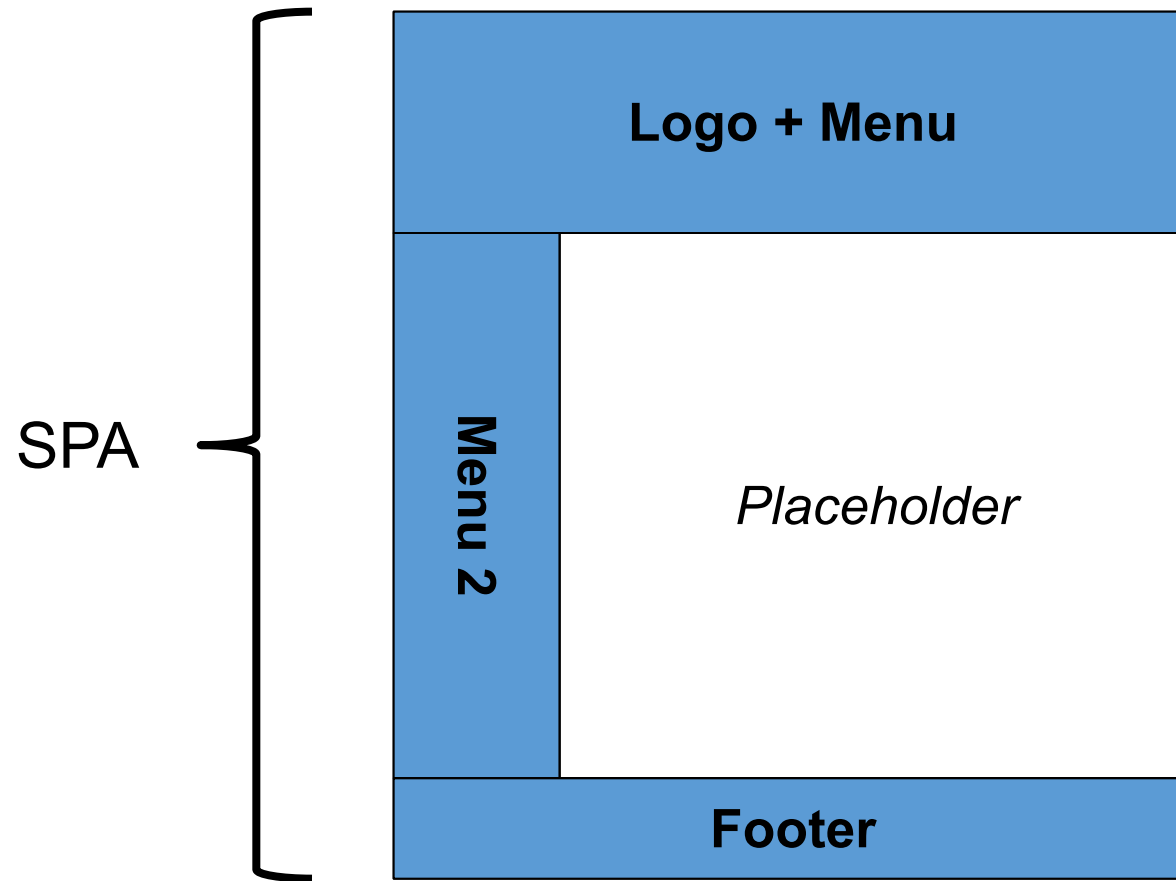


# Motivation

- SPAs → single page application
- Simulate pages → routes
- URL should direct to the routed component
  - Menus & bookmarks (today → Google 😊)
  - Sharing (social platforms)
  - **The Back button!**
  - **For development**

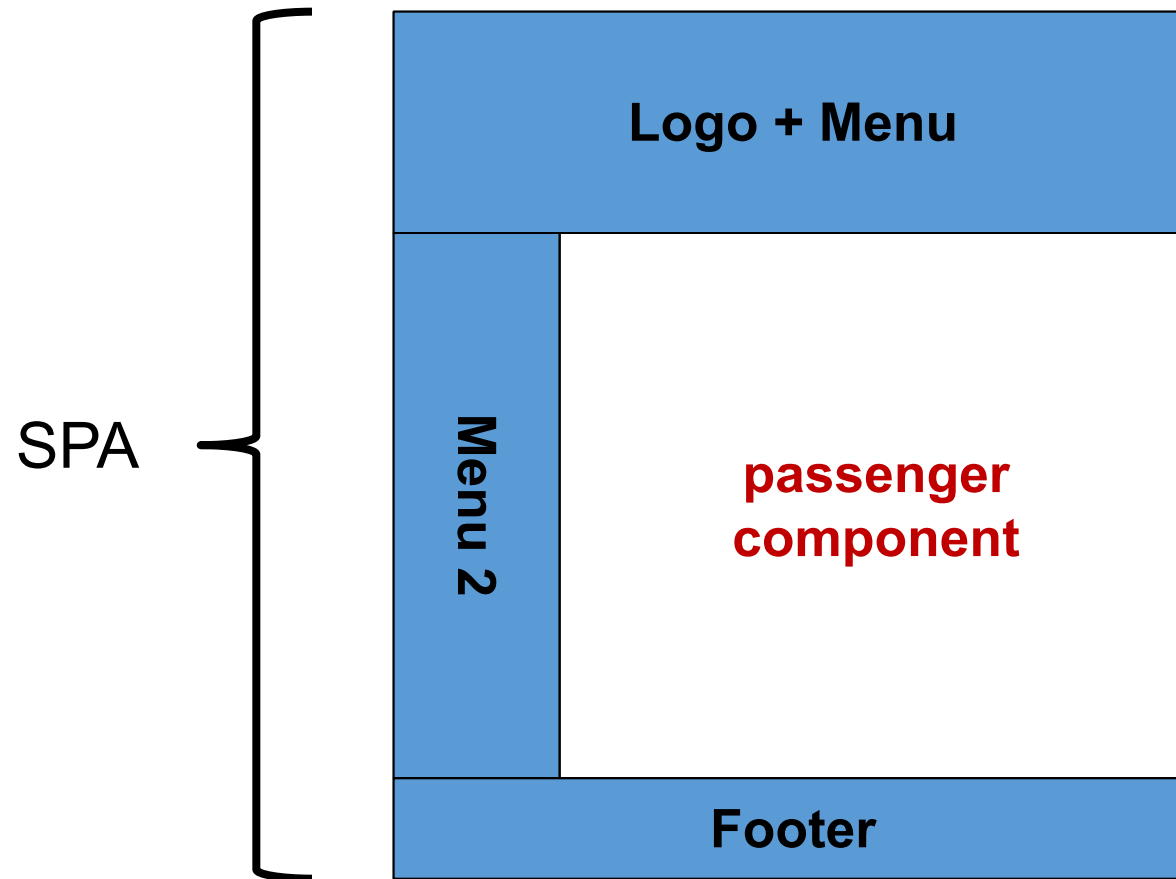


# Routing in Angular



# Routing in Angular

/flight-demo/**passenger**



# Routing via hash fragment

- /flight-demo#passenger
- Hash-Fragment is never sent to the server
- But we don't like this, do we?

# Routing via History API

- /flight-demo/**passenger**
- Readable for humans!
- Whole URL will be send to the server
  - Server responds with the SPA
  - Server can optionally render the initial view
    - Performance, SEO, ...
  - SPA informs the browser which part of the URL belongs:
    - to the server
    - to the Angular Route







# Angular Router



# Preparation

- Angular Module: RouterModule (@angular/router)
- 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: 'page-not-found'  
  }  
]
```

←----- **Last route!!!**



# Configuration

```
// app.module.ts
@NgModule({
  imports: [
    BrowserModule,
    HttpClientModule,
    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>
```



# Programmatic routing

```
export class AppComponent {  
    constructor(private router: Router) {}  
  
    navigateToFlightSearch(): void {  
        this.router.navigate(['/flight-search']).then();  
    }  
}
```



# Routing with parameters

# Parameters

- flight-edit/7
- flight-edit/7;showDetails=true;page=7
- flight-edit/7;showDetails=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 {  
  id = '';  
  
  constructor(private route: ActivatedRoute) {  
  
    ngOnInit(): void {  
      this.route.params.subscribe(  
        (params) => {  
          this.id = params['id'];  
          this.showDetails = params['showDetails'];  
          [...]  
        }  
      );  
    }  
    [...]  
  }  
}
```





# Parse parameters - alternative

```
export class FlightEditComponent {  
  id = '';  
  
  constructor(private route: ActivatedRoute) {  
  
    ngOnInit(): void {  
      this.route.paramMap.subscribe(  
        (paramMap) => {  
          this.id = paramMap.get('id');  
          this.showDetails = paramMap.get('showDetails');  
          [...]  
        }  
      );  
    }  
    [...]  
  }  
}
```



# Parse parameters - number

```
export class FlightEditComponent {  
  id = 0;  
  
  constructor(private route: ActivatedRoute) {  
  
    ngOnInit(): void {  
      this.route.paramMap.subscribe(  
        (paramMap) => {  
          this.id = Number(paramMap.get('id'));  
          this.showDetails = !!paramMap.get('showDetails');  
          [...]  
        }  
      );  
    }  
    [...]  
  }  
}
```



# Links of routes with parameters

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

# Links of routes with parameters

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



# DEMO



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# LAB



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**