



Navigation und Routing

Hosted by Alex Thalhammer

Inhalt

- Motivation
- Konfiguration
- Routing-Parameter

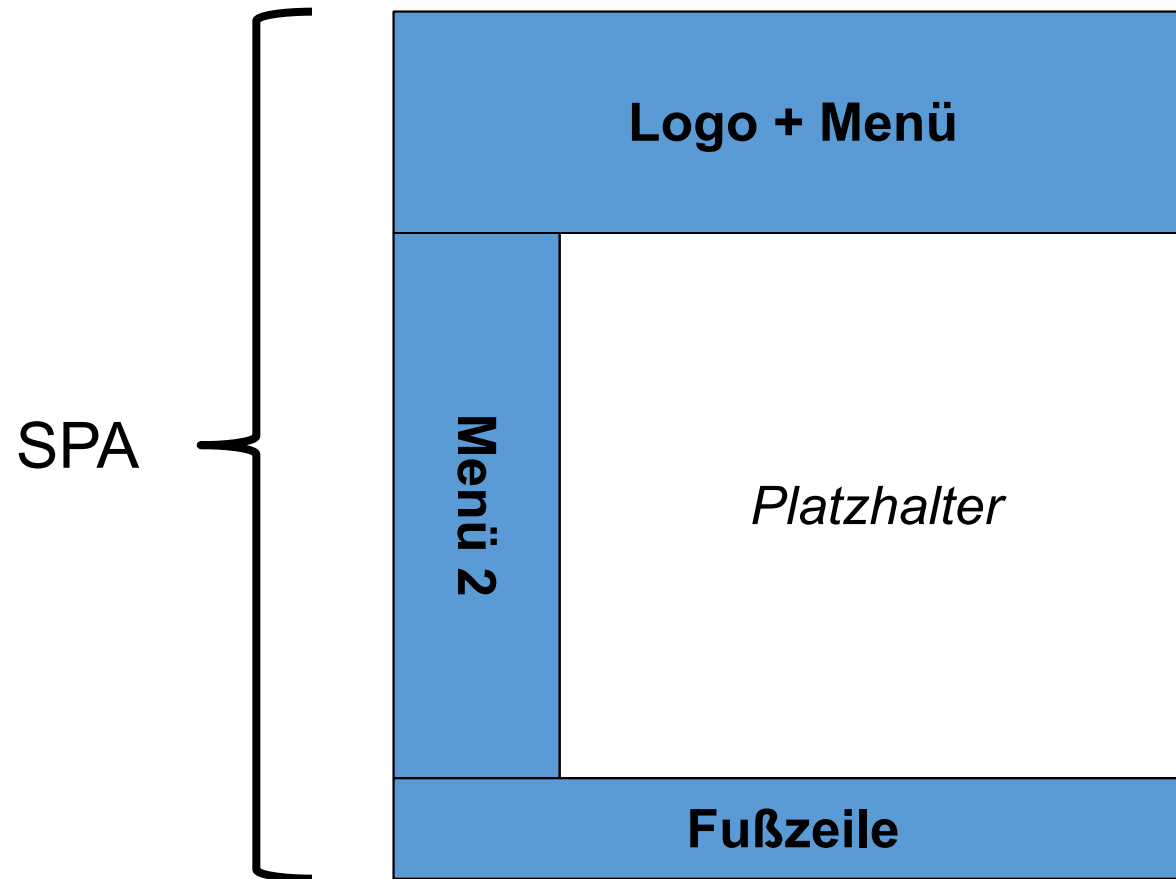


Motivation

- SPAs bestehen aus einer Seite
- Seiten simulieren → Routen
- Url sollte auf Route hinweisen
 - Bookmarks
 - Back-Button
- Router unterstützen hierbei

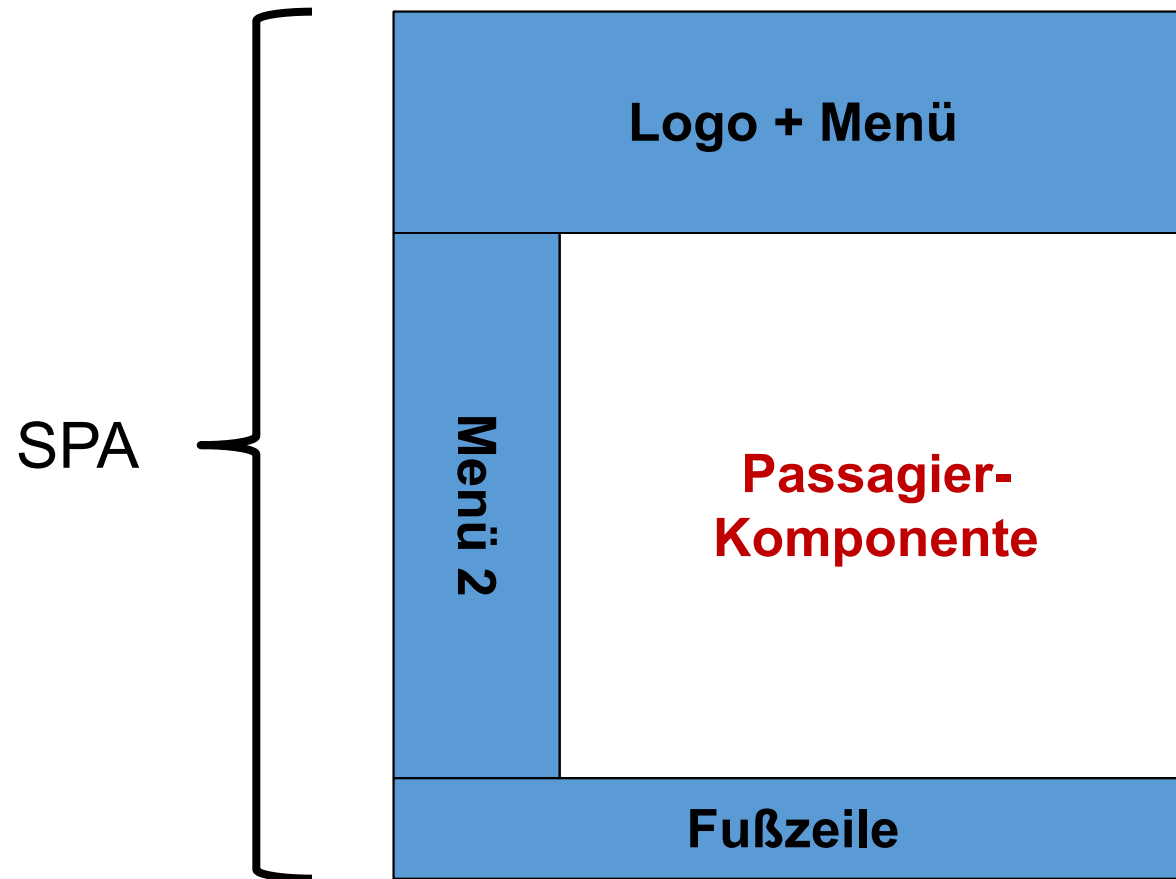


Routing unter Angular



Routing unter Angular

/FlugDemo/**passagier**





Angular Router

Konfiguration

```
export const APP_ROUTES: Routes = [  
  {  
    path: 'home',  
    component: HomeComponent  
  },  
  {  
    path: 'flug-suchen',  
    component: FlugSuchenComponent  
  }  
]
```



Konfiguration

```
export const APP_ROUTES: Routes = [  
  {  
    path: '',  
    redirectTo: 'home',  
    pathMatch: 'full'  
  },  
  {  
    path: 'home',  
    component: HomeComponent  
  },  
  {  
    path: 'flug-suchen',  
    component: FlugSuchenComponent  
  }  
]
```



Konfiguration

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



Konfiguration

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

Für Root-Module

Für Feature-Module: forChild



View von AppComponent

```
<a [routerLink]=" '/' ">Home</a>  
<a [routerLink]=" '/flug-suchen' ">Flug suchen</a>  
  
<div>  
  <router-outlet></router-outlet>  
</div>
```



View von AppComponent

```
<a routerLink="/">Home</a>  
<a routerLink="/flug-suchen">Flug suchen</a>  
  
<div>  
  <router-outlet></router-outlet>  
</div>
```



Programmatisch routen

```
export class AppComponent {  
  
    constructor(private router: Router) {  
    }  
  
    navigateToFlugSuchen() {  
        this.router.navigate(['/flug-suchen']);  
    }  
  
}
```



DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



Routen mit Parameter

Parameter

- `passagier/7`
- `passagier/7?details=true&page=7`
- `passagier/7;details=true;page=7`
- `passagier/7;details=true;page=7/fluege`

Konfiguration

```
const APP_ROUTES: Routes = [  
  [...]  
  {  
    path: 'flug-suchen',  
    component: FlugSuchenComponent  
  },  
  {  
    path: 'flug-edit/:id',  
    component: FlugEditComponent  
  }  
]
```



Parameter auslesen

```
export class FlugEditComponent {  
  
    public id: string;  
  
    constructor(  
        private route: ActivatedRoute) {  
  
        route.params.subscribe(  
            p => {  
                this.id = p['id'];  
                [...]  
            }  
        );  
    }  
    [...]  
}
```



Links auf Routen mit Parameter

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



Links auf Routen mit Parameter

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

DEMO





Lazy Loading

Warum Lazy Loading?

- Module erst bei Bedarf nachladen
- Verbesserung der Start-Performance



Root Module mit 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)  
  }  
];
```


Routen für Feature Module

```
const FLUG_ROUTES = [
  {
    path: 'subroute',
    component: FlugBuchenComponent,
    [...]
  },
  [...]
]
```

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



Konfiguration von Webpack

- @ngtools/webpack loader

DEMO





Preloading

Idee

- Eventuell später benötigte Module werden mit freien Ressourcen vorgeladen
- Wird das Modul später tatsächlich benötigt, steht es augenblicklich zur Verfügung



Preloading aktivieren

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



DEMO

