Cross Cutting Concerns

Alex Thalhammer

Contents

- Router Guards
- HttpInterceptor
- Directives

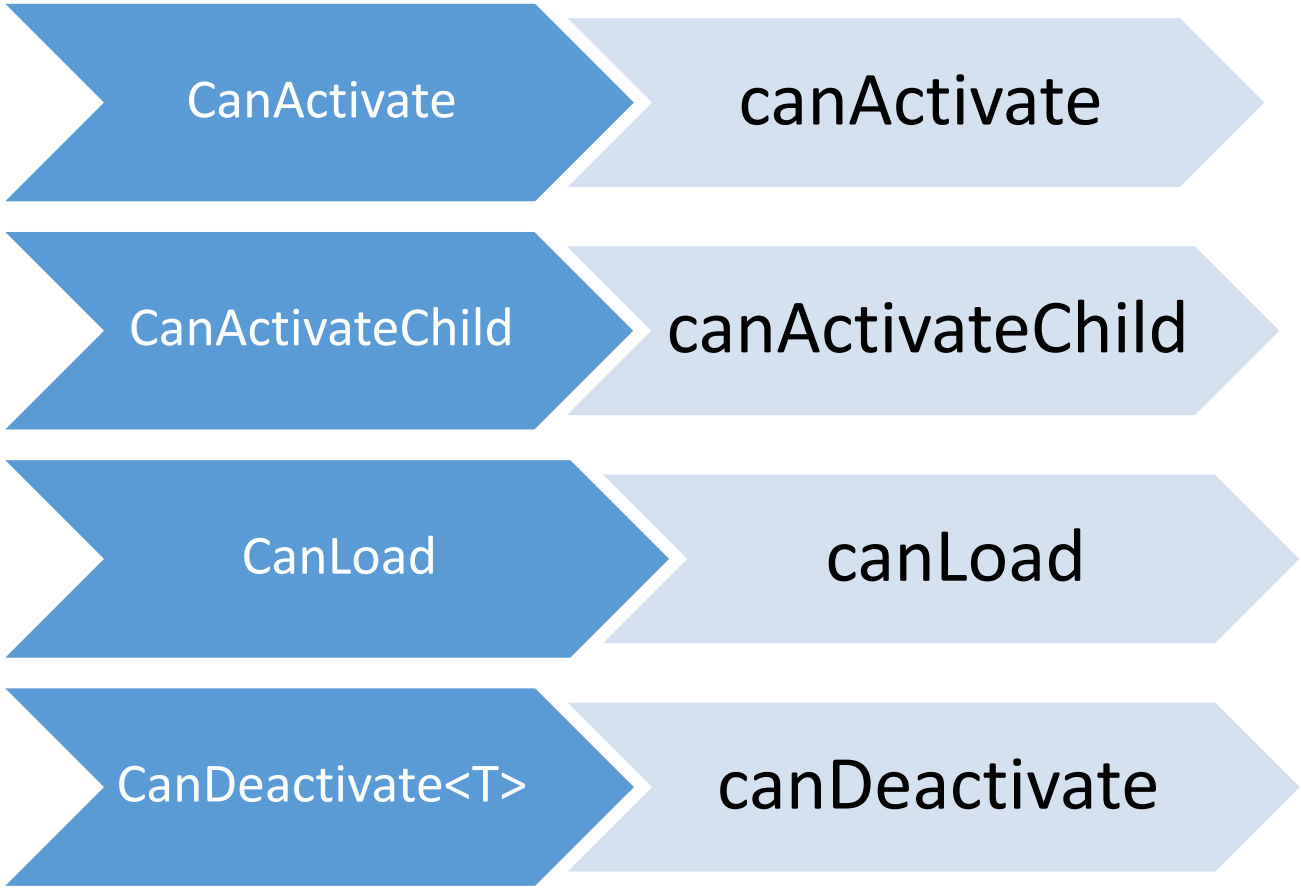
Router Guards

What are Guards?

- Services
- Can prevent the Activation or Deactivation of a Route

ANGULAR ARCHITECTS INSIDE KNOWLEDGE
SOFTWARE ARCHITECT

Guards



Result: boolean | Observable<boolean> | Promise<boolean>

Configure Guards

```
const APP_ROUTES: Routes = [
    {
        path: '/flight-booking',
        component: FlightBookingComponent,
        canActivate: [AuthGuard],
        children: [
            {
                path: 'flight-edit/:id',
                component: FlightEditComponent,
                canDeactivate: [FlightEditGuard]
            },
            […]
        ]
]
```


ANGULAR ARCHITECTS INSIDE KNOWLEDGE
SOFTWARE ARCHITECT

Provider for Guards

```
// app.module.ts
@NgModule({
    providers: [
        FlightEditGuard,
        AuthGuard
    ],
    […]
})
export class XYModule {
}
```

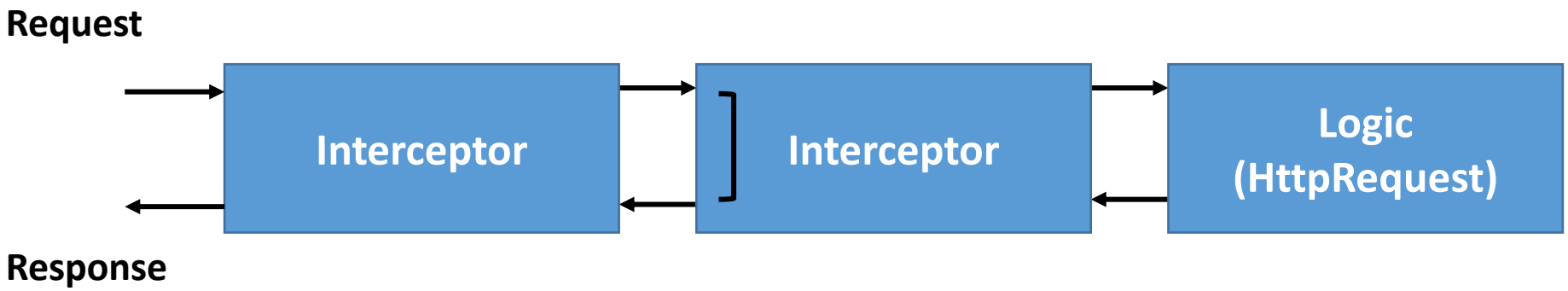

ANGULAR ARCHITECTS INSIDE KNOWLEDGE
SOFTWARE ARCHITECT

Http-Interceptors

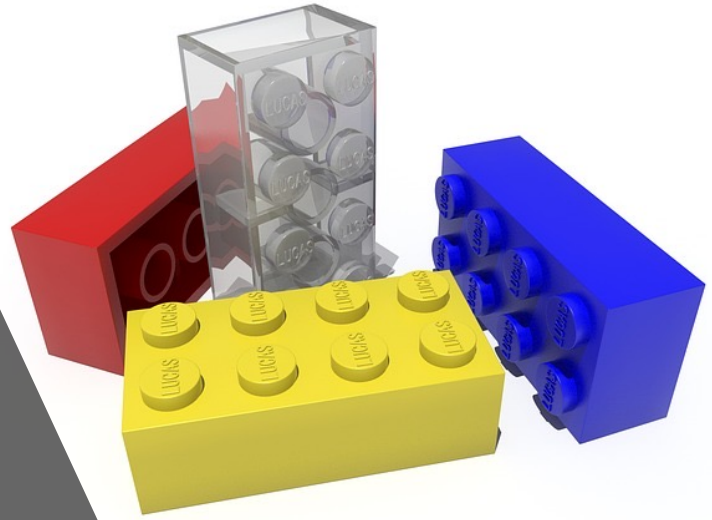
Idea

- Providing a **hook** for HttpClient
- Modify requests and responses globally
- Including Headers, e. g. for Auth.
- Parsing Data Formats, like XML
- Error Handling
- Caching
- …

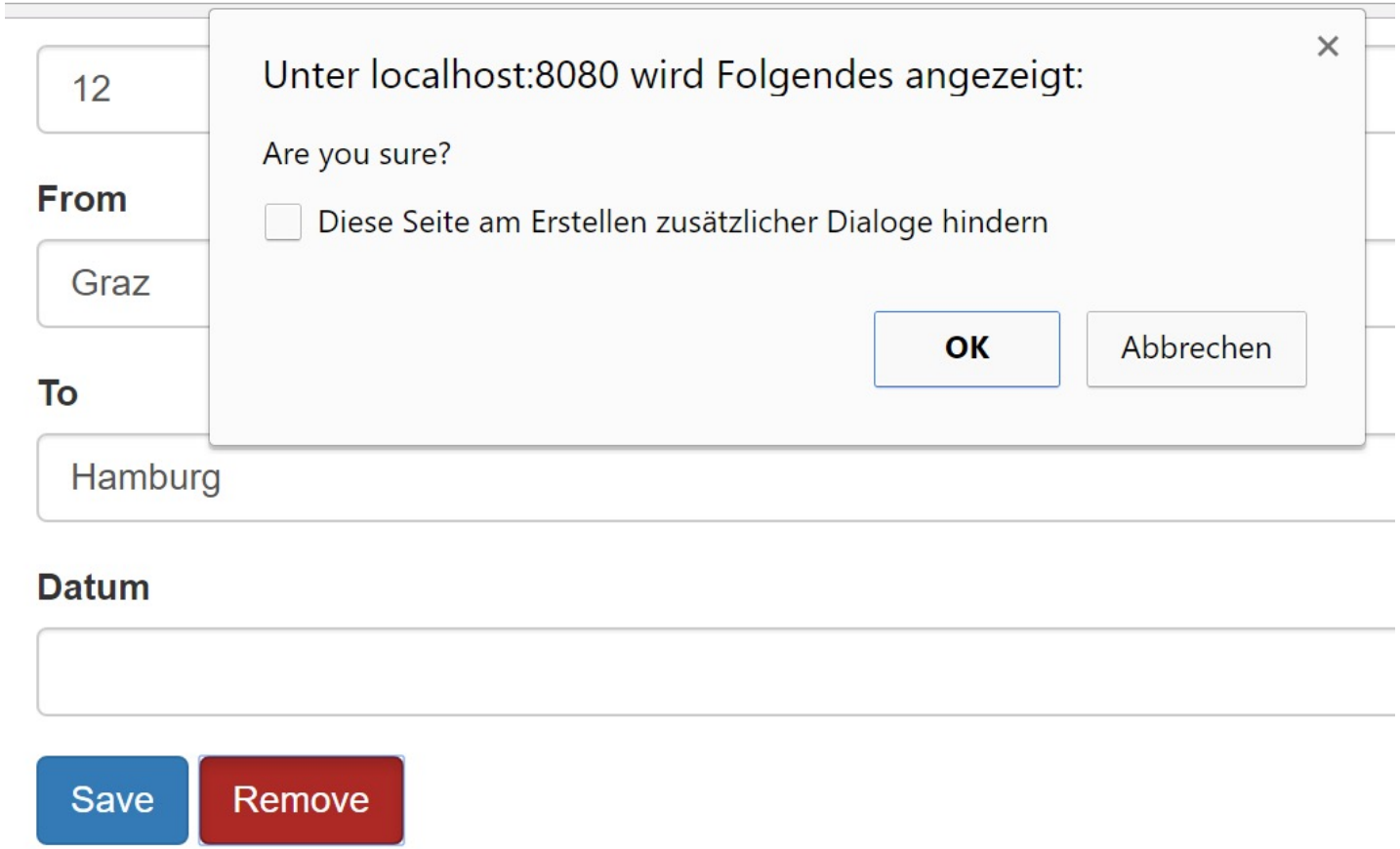
Chain of Responsibility



Directives



Warning Directive



Goal

<button **(clickWithWarning)="delete()" warning="really?"**>

Directive

```
@Directive({
    selector: '[clickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {

    […]

}
```

Directive

```
@Directive({
    selector: '[clickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {

    @Input() warning: string = 'Are you sure?';
    @Output() clickWithWarning = new EventEmitter();

    […]
}
```

Directive

```
@Directive({
    selector: '[clickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {

    @Input() warning: string = 'Are you sure?';
    @Output() clickWithWarning = new EventEmitter();

    @HostBinding('class') cssClass: string = 'btn btn-danger';

    […]
}
```

Directive

```
@Directive({
    selector: '[clickWithWarning]'
})
export class ClickWithWarningDirective implements OnInit {

    @Input() warning: string = 'Are you sure?';
    @Output() clickWithWarning = new EventEmitter();

    @HostBinding('class') cssClass: string = 'btn btn-danger';

    @HostListener('click', ['$event'])
    handleClick($event): void {
        if (confirm(this.warning)) {
            this.clickWithWarning.emit();
        }
    }
}
```

Lab

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT