



Services und Injection

Hosted by Alex Thalhammer

Was sind Services?

Wiederverwendbare
Codeeinheiten

Austauschbar

Testbar

Meist Klassen

Beispiel: FlugService



Service

Globaler Scope

```
@Injectable({ providedIn: 'root' })  
export class FlugService {  
  
    [...]  
  
}
```

**Services sind Singletons
(in ihrem "Scope")**



Alternative

Service wird mit lazy Module mitgeladen

```
@Injectable({ providedIn: LazyApiModule })  
export class FlugService {  
  
    [...]  
  
}
```

Macht nur mit Lazy Loading sinn!!

Details später in Kapitel "Routing"



Service

```
@Injectable({ providedIn: 'root' })  
export class FlugService {  
  
    [...]  
  
}
```



Konsument bekommt Service injiziert

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

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

  constructor(flugService: FlugService) { ... }

  flugSuchen() { [...] }
  selectFlug(flug) { [...] }
}
```

DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Konsument bekommt Service injiziert

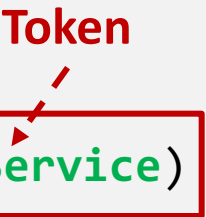
```
@Component({
  selector: 'flug-suchen',
  templateUrl: 'flug-suchen.html'
})
export class FlugSuchenComponent {

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

  constructor(flugService: FlugService) { ... }

  flugSuchen() { [...] }
  selectFlug(flug) { [...] }
}
```

Token



Token vs. Service

- Token: Das, was Konsument anfordert
(z. B. FlugService)
- Service: Das, was Konsument wirklich bekommt
(z. B. AdvancedFlugService)



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Token

- Fast alles kann ein Token sein!
- Häufig: Standard-Implementierung des Services
- Abstrakte (Basis)-Klasse
- Konstante
- NICHT: Interface

Weiterleitung

```
export abstract class FlugService {  
    abstract find(from: string, to: string): Observable<Flight[]>;  
}
```

```
@Injectable()  
export class DefaultFlugService implements FlugService {  
    find(from: string, to: string): Observable<Flight[]> { ... }  
}
```



Weiterleitung

```
@NgModule({  
  imports: [  
    BrowserModule, HttpClientModule, FormsModule  
  ],  
  declarations: [  
    AppComponent, FlugSuchenComponent  
  ],  
  providers: [  
    { provide: FlugService, useClass: DefaultFlugService }  
  ],  
  bootstrap: [  
    AppComponent  
  ]  
})  
export class AppModule {  
}
```



Weiterleitung

```
@NgModule({
  imports: [
    BrowserModule, HttpClientModule, FormsModule
  ],
  declarations: [
    AppComponent, FlugSuchenComponent
  ],
  providers: [
    { provide: FlugService, useClass: DefaultFlugService }
  ],
  bootstrap: [
    AppComponent
  ]
})
export class AppModule {
}
```



Weiterleitung

```
@NgModule({
  imports: [
    BrowserModule, HttpClientModule, FormsModule
  ],
  declarations: [
    AppComponent, FlugSuchenComponent
  ],
  providers: [
    { provide: FlugService, useClass: DefaultFlugService }
  ],
  bootstrap: [
    AppComponent
  ]
})
export class AppModule {
}
```

Token ↑

Service ↑



Weiterleitung

```
@NgModule({  
  imports: [  
    BrowserModule, HttpClientModule, FormsModule  
  ],  
  declarations: [  
    AppComponent, FlugSuchenComponent  
  ],  
  providers: [  
    { provide: FlugService, useClass: DefaultFlugService }  
  ],  
  bootstrap: [  
    AppComponent  
  ]  
})  
export class AppModule {  
}
```



Weiterleitung

```
@NgModule({  
  imports: [  
    BrowserModule, HttpClientModule, FormsModule  
  ],  
  declarations: [  
    AppComponent, FlugSuchenComponent  
  ],  
  providers: [  
    { provide: FlugService, useClass: FlugService }  
  ],  
  bootstrap: [  
    AppComponent  
  ]  
})  
export class AppModule {  
}
```



Weiterleitung

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



DEMO




ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Weiterleitung („neue“ Schreibweise)

```
@Injectable({  
  providedIn: 'root',  
  useClass: DefaultFlugService, ← -- Service  
  deps: [HttpClient]  
})  
export class FlugService {  
  [...]  
    
  Token  
}
```

Beispiel mit abstrakter Klasse

```
@Injectable({  
  providedIn: 'root',  
  useClass: DefaultFlugService  
})  
export abstract class FlugService {  
  
  abstract find(from: string, to: string);  
  
}
```

```
@Injectable()  
export class DefaultFlugService implements FlugService {  
  
  find(from: string, to: string) { ... }  
  
}
```



DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Service lokal registrieren

```
@Component({  
  selector: 'flug-suchen',  
  templateUrl: 'app/flug-buchen/flug-buchen.html',  
  providers: [{ provide: FlugService, useClass: FlugService}]  
})  
export class FlugSuchenComponent {  
  
  [...]  
  
}
```

Gilt nur für aktuelle Komponente und darunter!

Service lokal registrieren

```
@Component({  
  selector: 'flug-suchen',  
  templateUrl: 'app/flug-buchen/flug-buchen.html',  
  providers: [FlugService]  
})  
export class FlugSuchenComponent {  
  
  [...]  
  
}
```



DEMO

