



Services und Injection

Hosted by Alex Thalhammer

What are Services?

Reusable

Replaceable

Testable

Classes

Example:
FlugService



Service

global scope

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

**services are singletons
(in their “scope”)**



Alternative

Service is loaded with module

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

Makes only sense with Lazy Loading!

Details, later in chapter „Routing“



Service

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



Consumer gets injected service in constructor

```
@Component({
  selector: 'flight-search',
  templateUrl: 'flight-search.html'
})
export class FlightSearchComponent {

  from: string;
  to: string;
  flights: Flight[];

  constructor(flightService: FlightService) { ... }

  search(): void { [...] }
  select(flight): void { [...] }
}
```



DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Consumer gets injected service in constructor

```
@Component({
  selector: 'flight-search',
  templateUrl: 'flight-search.html'
})
export class FlightSearchComponent {

  from: string;
  to: string;
  flights: Flight[];

  constructor(flightService: FlightService) { ... }

  search(): void { [...] }
  select(flight): void { [...] }
}
```

Token



Token vs. Service

- Token: What the consumer requests
(e. g. flightService)
- Service: What the consumer receives
(e. g. advancedFlightService)



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE




SOFTWARE
ARCHITECT

Token

- Almost everything can be a token
- Most often: default implementation of service
- Abstract (Base)-Class
- Constant
- But no interface



Redirection with useClass

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

Example with abstract class

```
@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

Register service locally

```
@Component({  
  selector: 'flight-search',  
  templateUrl: flight-search.html',  
  providers: [FlightService]  
})  
export class FlightSearchComponent {  
  
  [...]  
  
}
```



Register service locally w. useClass

```
@Component({
  selector: 'flight-search',
  templateUrl: 'flight-search.html',
  providers: [{ provide: FlightService, useClass: AdvancedFlightService}]
})
export class FlightSearchComponent {

  [...]

}
```

Only used for current component and children

DEMO



One more thing: Angular Dev Tools

- <https://chrome.google.com/webstore/detail/angular-devtools/ienfalfjdbdpebioblackkekamfmbnh>