

# Spickzettel Routing

## Service

---

- Model `Book` ändern: jedes Buch bekommt beim Erstellen automatisch eine eindeutige ID
- Ordner `services` anlegen
- Service erstellen: `ng g service services/BookStore`
- drei Methoden:
  - `getBook(id)`: liefert das Buch mit der ID `id` zurück
  - `getAll()`: liefert alle Bücher als Array zurück
  - `addBook(book)`: fügt ein neues Buch in die Liste ein
- Provider registrieren
  - Wichtig: in `AppComponent`, damit Service-Instanz in den Kind-Komponenten von `AppComponent` erhalten bleibt
  - `providers: [BookStoreService]`
- Service einbinden in `DashboardComponent`
  - injizieren: `private bs: BookStoreService`
  - `getAll()` aufrufen in `ngOnInit` und beim Hinzufügen eines Buchs
  - `addBook()` aufrufen beim Erstellen eines Buchs. `Book`-Objekt übergeben

## Routing

---

- Globale Routenkonfiguration
  - `app.routes.ts` anlegen
  - Objekt von `RouterConfig` exportieren
  - darin Routen anlegen
    - `{ path: '', component: DashboardComponent }`
    - Bedeutung: der Pfad `/` lädt die Komponente `DashboardComponent`
    - Pfade tragen niemals einen Slash davor!
- `RouterConfig` in Anwendung registrieren
  - `main.ts`
  - Provider beim Bootstrapping registrieren
  - `provideRouter()` erhält Routenkonfiguration aus `app.routes.ts`
  - `bootstrap(AppComponent, [ provideRouter(AppRoutes) ])`
- `RouterOutlet` in `AppComponent` einbinden
  - `ROUTER_DIRECTIVES` importieren
  - `<router-outlet></router-outlet>` einbinden (statt `Dashboard`-Komponente)
- Fertig
- Beim Aufruf der Seite wird das Dashboard unter dem Pfad `/` geladen

## Detail-Komponente

---

- `ng g component BookDetails`
- Route anlegen
  - `{ path: 'book/:id', component: BookDetailsComponent }`
  - Pfad `book/:id` lädt die Komponente `BookDetailsComponent`
  - `:id` ist Platzhalter für Parameter
- Verlinken
  - in `BookComponent` Button mit Link anlegen
  - `ROUTER_DIRECTIVES` importieren
  - als Parameter die ID des Buchs übergeben
  - `<a [routerLink]='["book", book?.id]">Anzeigen</a>`
- Ausprobieren
  - Wird Button angeklickt, leitet Anwendung weiter zur (leeren) Detailseite
  - ID steht in der URL
- Parameter auslesen
  - `ActivatedRoute` injizieren (in `BookDetailsComponent`): `private route: ActivatedRoute`
  - `ActivatedRoute.params` ist `Observable`!
  - subscriben und ID auslesen:
    - `this.route.params.subscribe(params => { let id = params['id'] });`
  - zur Demonstration die ID in Konsole loggen?
- Buch über Service abrufen
  - in Handler-Funktion für Subscription: `this.book = this.bs.getBook(id)`
  - zur Demonstration das ganze Buch auf Konsole loggen oder auf Seite anzeigen: `{{ book | json }}`
- Template anlegen
  - von `BookComponent` kopieren, aber Rating entfernen