

## Arbeitsblatt MapStruct

### Ziele

Ziel dieses Arbeitsblattes ist es, dass Sie sich mit der MapStruct Bibliothek vertraut machen mit welcher Entities in DTOs kopiert werden können.

### Ausgangslage

Im Gitlab-Projekt <https://gitlab.fhnw.ch/eaf/hs21/05> steht das Projekt lab-mapstruct zur Verfügung, welches für diese Aufgaben verwendet werden soll. Importieren Sie dieses Projekt in ihre IDE.

Dieses Projekt definiert die beiden Domänenobjekte User und Rental sowie die DTOs UserDto und RentalDto. Mit Hilfe von *MapStruct* werden die Domänenobjekte in die entsprechenden DTOs kopiert.

```
public class User {
    private Long id;
    private String firstName, lastName;
    private Set<Rental> rentals = new HashSet<>();

    // Getter and Setter for all properties
}

public class UserDto {
    private long id;
    private String firstName, lastName;
    private Set<Long> rentals = new HashSet<>();

    // Getter and Setter for all properties
}
```

Der Unterschied zwischen diesen beiden Klassen liegt darin, wie die Referenzen auf die Ausleihen repräsentiert sind.

MapStruct verwendet APT (Annotation Processing Tool) um die Mapper-Klassen zu generieren. Sie müssen daher zuerst mit Gradle diese Klassen erzeugen, damit zur Laufzeit die Spring-Abhängigkeiten aufgelöst werden können. Führen Sie dazu den Befehl

```
gradlew build
```

aus. Diesen Befehl müssen Sie jeweils ausführen wenn Sie die Konfiguration (also das Mapper-Interface) geändert haben. Gradle ist so konfiguriert, dass MapStruct die Mapper-Klassen im Verzeichnis

```
build/generated/mapstruct
```

ablegt. Dieses Verzeichnis müssen Sie in ihrer IDE auf den Klassenpfad aufnehmen, d.h. z.B. als zusätzliches Source-Verzeichnis einrichten, sonst wird Spring die generierte Klasse nicht finden und mit dem Fehler

```
Field mapper in ch.fhnw.eaf.Main required a bean of type
'ch.fhnw.eaf.mappers.MovieMapper' that could not be found.
```

abbrechen. In der aktuellen Version habe ich versucht diesen zusätzlichen Pfad gerade in build.gradle zu definieren, aber ich weiss nicht ob es mit allen IDEs funktioniert.

APT bzw. MapStruct wird beim Erzeugen folgende Warnung ausgeben:

```
MovieMapper.java:17: warning: Unmapped target property: "userId".
    RentalDto rentalToRentalDto(Rental rental);
```

Dieses "Problem" werden Sie in Aufgabe 2 angehen und lösen.

### Aufgaben:

- 1) Studieren Sie anhand der Methode `Main.test1`, wie die Rental-Referenzen in einem User-Objekt auf die Rental-IDs bei einem `RentalDto` abgebildet werden.
- 2) In der Methode `Main.test2` wird die Referenz von der Rental-Instanz auf die User-Instanz nicht richtig abgebildet (im `RentalDto` steht dazu das Feld `userId` bereit). Wie kann erreicht werden, dass MapStruct dieses Feld ebenfalls setzt? Studieren Sie dazu Kapitel 5.3 der MapStruct Dokumentation auf der Seite <https://mapstruct.org/documentation/stable/reference/html/index.html#controlling-nested-bean-mappings>
- 3) Wir haben im Beispiel nur Entitäten in DTOs kopiert. Wie könnte man umgekehrt DTOs in Entitäten konvertieren damit diese dann in den Persistenzkontext eingefügt werden können?