

# Spring Framework

## Übung 3

Die Aufgabe war es, die in Übung 2 implementierten Daos durch Repositories der Spring Data JPA zu ersetzen. Für die Geschäftslogik wurde wie in der Übung ein Interface angelegt, welches anschließend Implementiert wurde. Die Konfigurationen für das Spring Framework wurde anhand von Annotationen in den Klassen vorgenommen.

Die einzelnen Repositories werden via „Autowire“ automatisch verlinkt. Ebenso werden Komponenten nicht in der XML Datei registriert, sondern ebenfalls anhand Spring Annotationen. Für das Custom Repository wurde ein Interface definiert, welches von dem Repository dann implementiert wird.

### Tests

Die Geschäftslogik wurde anhand 17 Testmethoden getestet.

Die für die Tests benötigten Daten werden direkt vorher eingefügt, darauf der Test ausgeführt, und dann ein Rollback auf die Datenbank abgesetzt. Damit ist jede Testmethode vor einer initialisierten Datenbank, und somit wird sichergestellt, dass die Methoden unabhängig voneinander durchgeführt werden.

Die Bonusaufgabe (Implementierung von UnitTests) wurde nicht erledigt.

### Hervorgekommene Probleme

Ein Problem welches ein Lazy-Fetch Exception erzeugt hat, war, dass die Geschäftslogik Ein Objekt geladen hat, jedoch inhalte von properties mittels Lazy fetch erst beim Zugriff nachgeladen hat. Wie in der Übung habe ich einen JPAInterceptor verwendet um dieses Problem zu beheben.

### UI

Ähnlich wie in der Übung 2 habe ich einen Konsolenanwendung geschrieben, bei dem Commands abgesetzt werden können. Diese rufen Methoden aus der BL auf, und geben die geforderten Informationen zurück.