

Aufgabe BookStore-Service

Schritt 1

Es kann als Basis auf dem Code aus dem Github-Projekt

https://github.com/Integrata-Cegos/Telekom_AgileAcademy/blob/02-03-Java/README.md

aufgesetzt werden. (Branch 02-03-Java)

Es handelt sich dabei um das begleitende Übungsprojekt (ist nicht von mir) einer anderen Trainee-Maßnahme. Und zu jedem Schritt gibt es exemplarisch eine Lösung im jeweiligen Branch.

Man muss sich den Test anschauen, um zu sehen, wie das Ganze funktioniert (es ist keine main-Methode vorhanden).

Das ist dann im ersten Schritt wie folgt zu erweitern/umzustellen:

- sprachlich aufzuräumen, z.B. Konstruktoren einführen, ...
- Einführung von Interfaces
- Unterscheidung zwischen api- und impl-Paket
- Einführung "richtiger" Tests:
 - BooksServiceOperationsTest: Testet alle Methoden
 - BooksServiceScenarioTest: Ausführung einer Test-Sequenz
 - BooksServiceFindBookByIsbnTest mit Hilfe von Mockito
- wenn man möchte kann man die Generators-Map auch gegen was einfacheres ersetzen

Wenn man sich die sprachliche Bereinigung sparen will, kann man auch auf der nächsten Stufe des Projekts aufsetzen:

https://github.com/Integrata-Cegos/Telekom_AgileAcademy/tree/04-Refactoring_Tests

und ist damit bei Schritt 2.

Schritt 2

Das ist dann wie folgt zu erweitern/umzustellen:

- Umstellung auf Spring Boot Projekt
- Umstellung auf Spring
 - Annotieren der Fach-Klassen mit `@Component`
 - Konfiguration mit `application.properties` und `@Value`
 - Optional: Einführung der Profiles `dev` (Starten der Application) und `test` (für die Spring Boot Tests)
 - Einführen einer Konfigurationsklasse für die Erzeugung der diversen Maps (books, store, generators)
- Umstellung der Tests auf Spring Boot Tests mit Nutzung von Jupiter (JUnit5)
 - Optional: Eigene Bean-Konfiguration für die Tests
 - Nutzen der Spring-Möglichkeiten (DI) im Test

Schritt 3

Das ist dann wie folgt zu erweitern/umzustellen:

- Umstellung auf Datenhaltung in einer relationalen Datenbank
 - DatabasesbnGenerator mit JPA native Queries
 - `JpaStoreService` mit native Query für Stock, Entity für Speichern
 - `JpaBooksService` mit Verwendung eines Spring-Data-Repositories

Hinweise

- Sowohl die Tests als auch die Applikation kann mit der Embedded Datenbank aber auch mit einer externen H2 gemacht werden. Beispieldaten müssen natürlich erstmal erzeugt werden.
- Die ganzen List-Methoden im Booksservice können als Default-Methoden ins Interface gezogen werden oder aber die ganze Sortiererei etc. sollte viel besser die Datenbank machen. Das wäre ein Beispiel für die Erweiterung des Spring-Data-Repositories.

Schritt 4

Das ist dann wie folgt zu erweitern/umzustellen:

- RESTful Facaden für alle Services
- CRUD für BooksService
- Lesen und Schreiben/Neuanlegen von Store-Einträgen mit POST
- Generieren einer ISBN mit POST
- Optional: Exception-Handling, Xml-Format, Controller-Test, ...