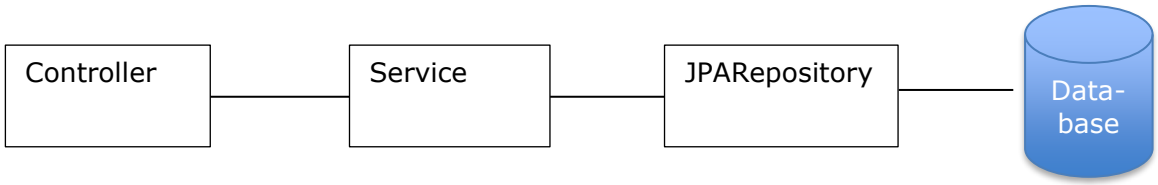
Unit-Test mit Mock und Datenbank

Einleitung:

Ausgangslage ist ein Backend, das mit Hilfe von einem Springboot Projekt den Zugriff auf eine Datenbank kapselt.



Der DatenbankService muss mit Unittests überprüft werden, ohne dass dazu die Datenbank zur Verfügung steht.

Die Idee ist es, das JPA-Repository durch einen Mock zu ersetzen und so die Tests unabhängig von der Datenbank zu machen.

Ziel

- ⇨ Sie können das Prinzip vom Mocken auf eine Situation mit einer Datenbank übertragen
- ⇨ Sie können Unittests mit Unterstützung eines Mocks für den Datenbankzugriff erstellen.

Inhalte

Einleitung: 1
Ziel 1
Inhalte 1
Einleitung 1
Aufgabe Beispiel untersuchen 1
Aufgabe Mock untersuchen 2

Einleitung

Ich habe mal die AI gefragt, wie man das Problem oben lösen könnte:

- Lesen Sie die Antwort

  20_8_ChatCPTPromtAndAnswer.pdf

Aufgabe Beispiel untersuchen

Sie finden das Beispiel von ChatGPT als Projekt umgesetzt in der Datei.
22_7_JPA-MyEntityExample.zip

- Entpacken Sie das Beispiel und öffnen Sie es mit Ihrer IDE.
- Untersuchen Sie das Beispiel

Fragen:

- Was passiert, wenn Sie die Applikation starten?
  Erklären Sie den Grund für das Verhalten.

- Wieso können Sie dennoch die Unittests ausführen?
  Erklären Sie den Grund.

⇨ Der Test *testFindByName_ReturnsNull_WhenNotFound* schlägt fehlt.
Dazu später mehr

Aufgabe Mock untersuchen

Arbeiten Sie zu zweit:

- Erklären Sie Ihrem Lernpartner, wieso es in der Klasse MyEntity keinen *Boiler platte code* wie Getter, Setter, Constructoren hat.
- Im MyEntityService finden Sie eine *Constructor injection*.
  Recherchieren Sie, was das ist.
  Erklären Sie es Ihrem Lernpartner
- Im MyEntityService gibt es die beiden Methoden:
  findById
  findByName
  Erklären Sie Ihrem Lernpartner, was diese Methoden machen.
- Erklären Sie wieso es im MyEntityRepository nur diese Methode hat:
  findByName
- Erklären Sie Ihrem Lernpartner, was ***Optional***<*MyEntity*> ist.

Teilen Sie sich auf, einer nimmt den Test testFindById, der andere den Test testFindById_ReturnsNull_WhenNotFound.

- Untersuchen Sie Ihren Test, so dass Sie Ihrem Lernpartner jeden Schritt im Test erklären können.
- Erklären Sie sich gegenseitig den Test.

- Was ist der Unterschied von @Mock und @InjectMocks?

Der Service macht doch eigentlich das Gleiche wie das Repository. Wieso braucht es den Service überhaupt, könnte man nicht direkt Unittests für das Repository schreiben?

- Diskutieren Sie diese Idee.

Der Test *testFindByName_ReturnsNull_WhenNotFound* schlägt fehlt.

- Erklären Sie wieso.
- Fixen Sie.. falls nötig.