

Übungen JDBC

0. Vorbereitung

Klonen Sie das Projekt und führen Sie es erfolgreich aus.

Nutzen Sie dafür die Anweisungen der [README.md](#).

1. Dateneingabe

- Füge eine SQL Anweisung hinzu, welche zufällige Nummern in die Datenbank hinzufügt
- Diese sollen durch Java Code produziert werden
- Generiere dann die Beschreibung einer Lieblingszahl zufällig

2. Integrationsbedingungen

- Füge eine Nummer nur dann hinzu, wenn diese noch nicht existiert
- Realisiere dies über eine SQL-Abfrage bevor du Daten hinzufügst
- Füge der `favorite_number` Tabelle nun eine Integrationsbedingung hinzu, sodass `number` einzigartig sein muss (eventuell musst du die DB löschen)

3. SQL-Injection

Mit SQL-Injection kann bösartiger Code in ihre Datenbank gelangen und großen Schaden anrichten. In dieser Übung simulieren Sie einen solchen Angriff.

- Schreibe eine neue Funktion, welche eine Lieblingszahl anhand der Begründung (`reason`) abfragen kann (bspw. `queryByReason(String reason)`)
- Nutzen eine einfache Abfrage wie `ResultSet rs = stmt.executeQuery("SELECT * FROM favorite_number WHERE reason LIKE '" + reason + "'");`
- Nutze die Funktion, um eine Zahl anhand der Begründung auszugeben
- Übergebe nun einen Parameter, welcher bösartiges SQL injiziert (bspw. `DROP TABLE favorite_number`)
- Schreibe eine zweite Funktion mit der gleichen Funktionalität, welche ein PreparedStatement nutzt und SQL Injection verhindert

4. Fehlerbehandlung

- Entferne die `throws SQLException` Anweisungen
- Handle die Fehler durch einen `try ... catch` Block
- Beachte das korrekte Schließen der verwendeten Ressourcen
- Füge eine fehlerhafte SQL Anweisung hinzu, um einen der `try ... catch` Blöcke auszulösen