



1. Erweitern Sie die Übung aus dem letzten Kapitel indem Sie die von der Web-Anwendung zu verwaltenden Daten in eine *MySQL-Datenbank* schreiben. Die Datenbankdefinition nebst Beispieldatensätzen finden Sie in den Rohdateien.

Verwenden Sie in den jeweiligen Modellen für den Zugriff auf die Datenbank ein Objekt der vorgestellten Database-Klasse.

Um das Passwort des Benutzers zu hashen sollen Sie das in Node.js vorhandene `crypto` Modul verwenden. Näheres dazu finden Sie unter <https://nodejs.org/en/knowledge/cryptography/how-to-use-crypto-module/>.

```
const crypto = require('crypto');
const hash = crypto.createHash('sha256')
  .update(password)
  .digest('hex');
```

Achten Sie auch darauf, dass beim momentanen Ausfallen des Datenbanksystems Node.js nicht abstürzt, die Web-Anwendung mit einer *benutzerfreundlichen Fehlermeldung* antwortet, und dass beim Wiedervorhandensein der Datenbankverbindung die Web-Anwendung ihre Arbeit wieder fortsetzen kann.

Meldung

Die Web-Anwendung ist momentan überlastet. Probieren Sie es später wieder...

[Wiederholen](#)

Kontrollieren Sie beim Eintragen oder Ändern eines Films, dass der *Filmtitel eindeutig* ist. Es dürfen nicht Filme mit denselben Titeln mehrmals eingetragen werden.

BEMERKUNG: Auf eine komfortable *Validierung* der Benutzereingaben kann wiederum verzichtet werden. Wenn Falscheingaben gemacht werden, dann muss nur gewährleistet werden, dass die Web-Anwendung nicht abstürzt, sondern weiterarbeitet.

2. Realisieren Sie dann eine *Importfunktion* für Filme, die nur *angemeldete Benutzer* aufrufen dürfen. Wie in Node.js das Hochladen von Dateien realisiert werden soll, wird im *Exkurs* mit dem Titel „Datei-Upload in Node.js“ beschrieben.

Nur Filme im nebenstehenden *JSON-Format* können importiert werden. Sollte das Format nicht passen, muss eine entsprechende Fehlermeldung angezeigt werden.

```
[
  { "title": "Hulk", "year": 2008 },
  { "title": "Avengers", "year": 2012 },
  { "title": "Spider-Man", "year": 2007 }
]
```

Alle importierten Filme erhalten als *Besitzer* (owner) den *angemeldeten Benutzer* und sind *nicht öffentlich*. Die Filme werden der Reihe nach durch *einzelne* SQL INSERT-Befehle in die Datenbank aufgenommen. Vor dem jeweiligen INSERT-Befehl wird kontrolliert, ob der Film nicht schon von irgendeinem Benutzer eingetragen wurde, dann wird er in die Tabelle aufgenommen. Sollte im Laufe des Einfügens von Filmen ein bereits vorhandener Filmtitel entdeckt werden, müssen alle bereits eingetragenen Filme wiederum entfernt werden (**TIPP:** ROLLBACK).

Zum Schluss soll angezeigt werden, ob alle Filme erfolgreich importiert wurden bzw. bei welchem (ersten) Filmtitel ein Duplikat erkannt wurde.

Meldung

Import meldet:

Falsches JSON-Format

Meldung

Import meldet:

Film mit Titel Hulk bereits vorhanden

Meldung

Import meldet:

Filme erfolgreich importiert