

# Multimedia Engineering II - Übung 3

---

Maximale Punktezahl: **27 Punkten** (17 aus dem Pflichtteil, 5/5 Optional)

**20 Punkte → 100 %**

## Zielstellung der Übungsaufgabe

In den vergangenen zwei Übungen wurde hauptsächlich der Umgang mit HTML5 und JavaScript geübt. Für den in der letzten Übung erstellten Sequenzeditor soll nun eine Persistenzschicht entwickelt werden.

Speichern Sie jede Aufgabe in eine einzelne Dateien/Ordner ab! Achten Sie auf die Struktur, welche in der Einführung kommuniziert wurde!

**Wichtiger Hinweis:** Sollten Sie absehbar Probleme oder Schwierigkeiten mit der Lösung einer Aufgabe oder einem Teil davon haben, kommen Sie bitte regelmäßig in die Übung. Wir sprechen dort die einzelnen Aufgaben auch noch einmal gemeinsam durch.

## Informationen und Links

- <http://docs.slimframework.com/>
- <http://expressjs.com/api.html>
- <http://pivotal.github.io/jasmine/>

## Aufgabe 1 (Pflichtteil) – 5 Punkte:

Erweitern Sie im ersten Schritt ihren Sequenzeditor aus der zweiten Übung um eine Liste nach CRUD, welche es ermöglicht, Sequenzen anzulegen, sie zu benennen und ggf. wieder zu löschen.

Sequence List:

- Meine Sequenz X
- Noch eine Sequenz X

Create New

Delete

None

None

Meine Sequenz

Noch eine Sequenz

Abbildung 1 - Mögliches Interface

Dieses Interface soll in der folgenden Aufgabe die Datenbank ansprechen und die Verwaltung der Sequenzen übernehmen.

## Aufgabe 2 (Pflichtteil) – 9 Punkte

Implementieren Sie im Anschluss in PHP oder JavaScript (nodeJS) eine API für den in der zweiten Übung erstellten Sequenzeditor in Verbindung mit dem in Aufgabe 1 implementiertem Interface. Sie können für den Editor entweder die im Plug-In gekapselte oder normale Version nutzen. In jedem Falle sollen die Zugriffe auf die API **nicht** in einem Plug-In integriert werden. Folgende Kriterien muss Ihre API erfüllen:

- REST-Schnittstelle
  - Versionsnummer
  - Nutzen von HTTP-Zustandscodes
  - Header für die Rückgabe → Content Type
  - HTTP-Operationen nach CRUD
  - Pagination
  - Rückgabe in JSON
- Abfrage der Inhalte nur auf Basis der E-Mail-Adresse - keine weitere Authentifizierung
- CRUD-Interface
  - Erzeugen, Laden, Speichern und Löschen von Sequenzen
  - Es sollte möglich sein, neben einer Liste auch einzelne Sequenzen abzufragen

Überlegen Sie sich eine sinnvolle Benennung der Routen nach den o.g. und in der Vorlesung aufgezeigten Kriterien.

Speichern Sie die Daten in einer MongoDB oder in MySQL. Auch in MySQL benötigen Sie kein relationales Modell sondern nur eine Tabelle:

- Id
- Data (JSON-Objekt der Sequenz)
- Email

Für die Aufrufe der API am Frond-End können Sie z.B. einfach jQuery-Ajax nutzen.

## Aufgabe 3 (Pflichtteil) – 3 Punkte

Erzeugen Sie für Ihre API eine Dokumentation. Die Dokumentation kann entweder mit einem Texteditor oder mit Hilfe eines Frameworks erzeugt werden. Sie sollte in jedem Fall für jede der API-Calls folgende Punkte umfassen:

- Input – Erwartete Eingabe
- Output – Korrekte Ausgabe
- Example Input – Beispiel
- Example Output –Beispiel
- Error – mögliche Fehlermeldung(en)

## Aufgabe 4 (Optional) – 5 Punkte

Erzeugen Sie für Ihre API eine kleine Testsuite mit Jasmine.

Die Testsuite muss nur für jeden API-Call einen gültigen und einen fehlerhaften Testfall umfassen. Überprüfen Sie ob die zurückgegebenen Ergebnisse der Erwartung entsprechen. Achten Sie darauf, dass die Testsuite gesondert in eine eigene Datei geschrieben wird.

## Aufgabe 5 (Optional) – 5 Punkte

Erweitern Sie Ihren Sequenzeditor um eine Elementliste. Diese Liste wird ebenso durch eine zweite Tabelle/Collection verwaltet. Die Erzeugten Elemente können einen Titel und ein Thumbnail haben.

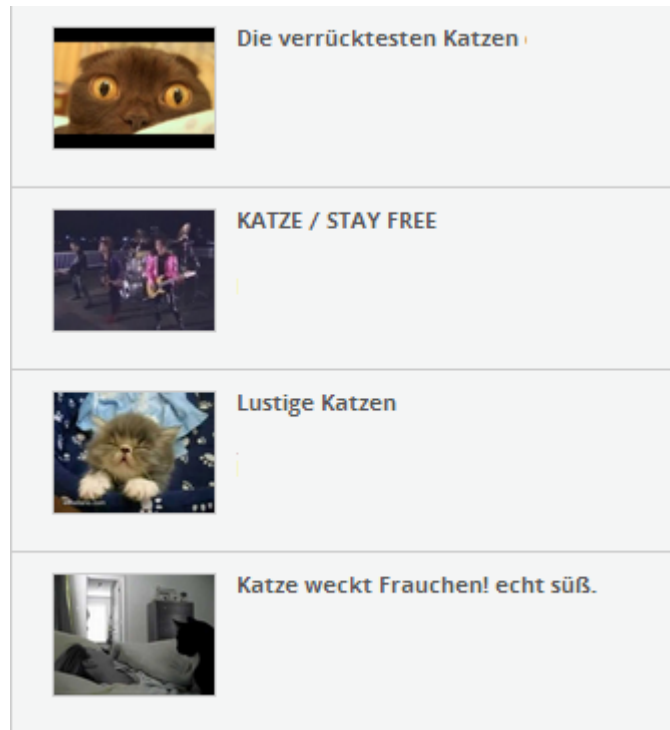


Abbildung 2 - Elementliste

Im Anschluss soll es möglich sein, diese Elemente auf dem Sequenzeditor einzufügen:



Abbildung 3 - Drag&Drop der Elemente auf die Editorfläche

Die Elemente sollen somit über Ihre Ankerpunkte verfügen und mit anderen verbunden werden können.