

# Übungsaufgabe Multimedia Engineering (MEI-M 04.3)

## MusicExplorer

2.12.2013

Abgabe 18.12.2013, 23:55 Uhr: nachname\_vorname\_musicexplorer.zip

Implementieren Sie eine Anwendung mit dem Framework **Twitter Bootstrap**, mit der Nutzer in der Musikplattform **SoundCloud** suchen und Tracks zu einer Playlist zusammenstellen können.

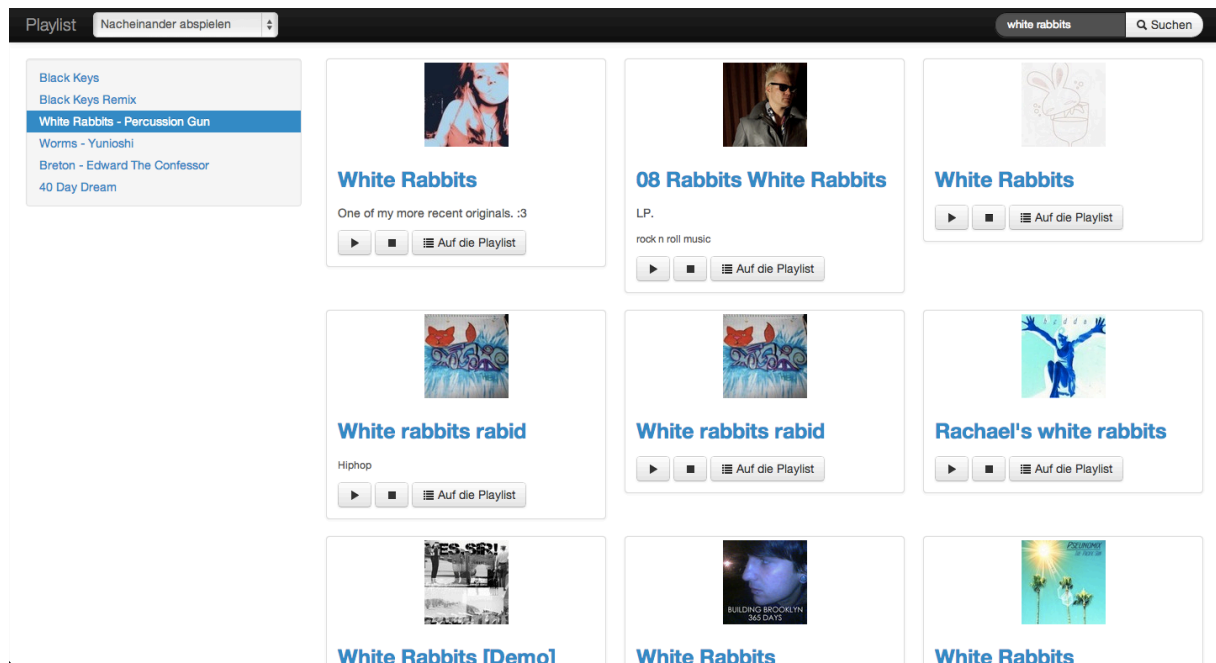


Abbildung 1: Hauptscreen der Anwendung mit Toolbar, Playlist und Suchauflistung der gefundenen Tracks.

### SoundCloud:

Um die Informationen aus der SoundCloud-API abrufen zu können, benötigen Sie zuerst eine sog. „Client ID“. Gehen Sie hierzu wie folgt vor:

1. Klicken Sie auf der Seite <http://developers.soundcloud.com/> rechts oben auf „Your Apps“.
2. Melden Sie sich mit Ihrem Facebook-Account an oder erstellen Sie ein neues Profil („Sign up“).
3. Nach dem Login klicken Sie im Bereich „Your applications“ auf „Register a new application“.
4. Vergeben Sie einen Namen und kopieren Sie sich dann den Schlüssel unter „Client ID“.
5. Fügen Sie die „Client ID“ in der Datei MainModel.js in der Methode SC.initialize ein.
6. Verschaffen Sie sich einen ersten Überblick über die Dokumentation des JavaScript SDKs (<http://developers.soundcloud.com/docs/api/sdks#javascript>) und der API-Dokumentation (insbesondere <http://developers.soundcloud.com/docs/api/reference#tracks>).

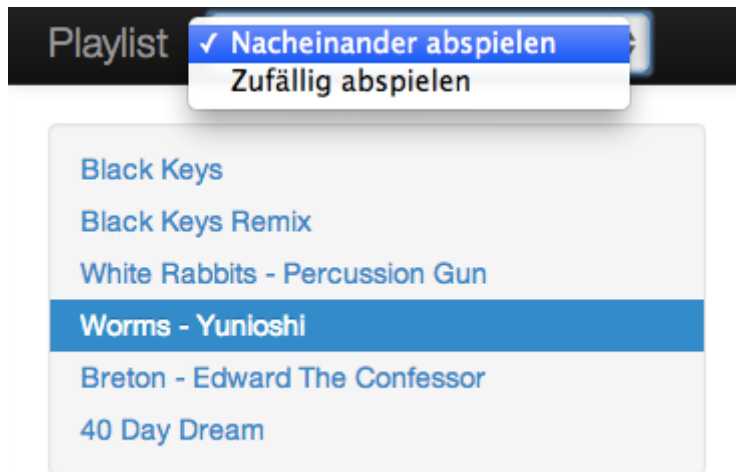


Abbildung 2: Playlist mit zwei wählbaren Modi („Nacheinander abspielen“ und „Zufällig abspielen“) und dem aktiven Track in der Liste. Die Tracks können in der Playlist auch direkt angewählt und abgespielt werden.



Abbildung 3: Einzelne Tracks der Suchauflistung. Links: Track mit Play-Button, Stop-Button und Playlist-Button. Rechts: Track, der bereits zur Playlist hinzugefügt wurde.

Folgende **Anforderungen** sind zu implementieren:

- In der Toolbar befindet sich rechts oben ein Eingabefeld und der Button „Suchen“. Wenn die Suche (mit Druck auf „Enter“ oder den Suchen-Button) gestartet wird, ändert sich der Titel des Suchen-Buttons in „Suche...“ und der Button wird deaktiviert. Wenn die Suche beendet ist, wird der Ursprungszustand des Buttons wieder angezeigt.
- Die Suchergebnisse werden mit drei Tracks pro Reihe angezeigt (siehe Abbildung 1). Jedes Suchergebnis zeigt einen User-Track an (siehe Abbildung 3), der angehört und auf die Playlist gesetzt werden kann.
- Jeder Track in der Suchauflistung stellt folgende Informationen dar (siehe Abbildung 3):
  - Das Bild des Tracks (artwork\_url) oder – falls nicht vorhanden – das Bild des Users (user.avatar\_url).
  - Den Titel des Tracks mit Link auf die SoundCloud-Seite (permalink\_url).
  - Eine Beschreibung des Tracks (description), sofern vorhanden.

- Einen Play-Button, der den Track abspielt (stream\_url). Wird ein neuer Track abgespielt, stoppt die Wiedergabe des aktuell spielenden Tracks zuerst.
  - Einen Stop-Button, der die Wiedergabe des Tracks stoppt.
  - Einen Playlist-Button, der den Track an das Ende der Playlist setzt. Sobald der Track hinzugefügt wurde, ändert sich der Titel des Buttons auf „Hinzugefügt!“ und der Button wird deaktiviert (siehe Abbildung 3 rechts).
- In der Toolbar kann links oben der Abspielmodus der Playlist über ein Auswahlménú auf einen von zwei verfügbaren Modi geändert werden.
  - Die Playlist startet die Wiedergabe bei dem Track, der angeklickt wurde. Der nächste Track ist entweder der direkt darauffolgende Track in der Liste (Modus „Nacheinander abspielen“) oder ein zufällig ermittelter Track in der Liste (Modus „Zufällig abspielen“). Im Modus „Nacheinanders abspielen“ wird nach dem letzten Track wieder der erste Track abgespielt. Zu jeder Zeit kann ein anderer Track ausgewählt werden, bei dem die Wiedergabe dann startet.
  - Hinweis: Es ist nicht erforderlich, dass die Playlist lokal gespeichert wird. Falls Sie eine derartige Funktion dennoch selbst implementieren wollen, finden Sie hier weitere Informationen hier: <http://diveintohtml5.info/storage.html>

Bitte beachten Sie folgende **Hinweise**:

- Die Anwendung muss von einem Webserver (http://...) gestartet werden, da sonst die Sound-Wiedergabe nicht funktioniert. Eine einfache Möglichkeit hierzu ist z.B. die Nutzung eines „Public Folders“ der Dropbox: <https://www.dropbox.com/help/16/en>. Eine Anleitung für einen lokalen Webserver mit XAMPP ist im GRIPS-Forum einsehbar.
- Verwenden Sie das Framework „**Twitter Bootstrap**“ (<http://getbootstrap.com/>) für das User Interface. Sehen Sie sich hierzu genau die Dokumentation und die Beispiele an.
- Setzen Sie die Anwendung konsequent nach dem **MVC-Pattern** um: Implementieren Sie SoundCloud-API-Methoden und Methoden zur Verwaltung der Playlist-Daten in Ihrem Datenmodell (MainModel.js). Implementieren Sie die Anzeige der Playlist im Modul PlaylistView und die Anzeige der Suchergebnisse im Modul TrackListView. Das Modul MainController koordiniert zwischen Datenmodell und den Views (Abrufen und Ändern der Model-Daten und aktualisieren der Views auf Basis der UI-Interaktion). Wichtig: Schreiben Sie keinen View-Code in Ihr Datenmodell!
- Nutzen Sie JavaScript-Templates (z.B. die Funktion `_.template` der **Underscore**-Bibliothek, siehe Übungsbeispiel) für die Einträge der Playlist und die Tracks der Suchauflistung.
- Versuchen Sie, das Design detailgetreu umzusetzen und orientieren Sie sich dabei an der Dokumentation des Twitter-Bootstrap-Frameworks, insbesondere am Examples-Bereich

(<http://getbootstrap.com/getting-started/#examples>) und dem Abschnitt über Thumbnails (<http://getbootstrap.com/components/#thumbnails>) im Components-Bereich. Einige Buttons verwenden Icons, die als Glyphicons (<http://getbootstrap.com/components/#glyphicons>) eingebunden werden können.

- Zusätzliche Features können gerne eingebaut werden. Vorstellbar ist z.B.
  - Speicherung der Playlist im Browser mittels LocalStorage
  - Entfernen eines Tracks aus der Playlist
  - Erkennung, ob ein Track bereits in der Playlist ist, auch nach erneuter Suche
  - Nachträgliche Änderung der Track-Reihenfolge
  - Weitere Music-Player-Funktionen (Pause, Next Track, ...)