

Dokumentation Java EE

SyncVideo

Das von uns erstellte Projekt besitzt den Namen SyncVideo. Die Idee hinter dem Projekt ist es, in verschiedenen Browsern und auf verschiedenen Computern (falls das Projekt auf einem Server installiert wurde auch an verschiedenen Orten) synchronisiert Videos anzuschauen. So lässt sich auch über weite Entfernung ein Filmabend gestalten.

Aufteilung der Arbeit

Fabian Iwanecki:

Erstellung des Frontends mittels JSP und Google Material Design; Implementierung des Videoplayers mit der Bibliothek VideoJs; Implementierung des Websocket für die Userliste, die Playlist, die Raumfunktionen und Synchronisierung von Videos (sowohl clientseitig als auch serverseitig); Mitarbeit bei der Persistenzschicht mittels JPA und JTA sowie den REST-Schnittstellen; Implementierung der YouTube-Videofunktion (synchronisierung mit anderen Nutzern funktioniert aufgrund einer Änderung in der Youtube-API nicht; Videos laden und spielen aber ab); Implementierung der Login-Funktionalität auf allen Schichten; Diverse Bugfixes in allen Bereichen der Anwendung

Niklas Jopke:

Erstellen und Verwalten der Persistence mittels JPA, MySQL, Hibernate, MySQL Connector und JDBC. Erstellen der Dokumentation (Bedienung und Einrichtung).

Diverse Bugfixes in allen Bereichen der Anwendung.

Christoph Kandler:

Erstellen der REST-Schnittstellen, Mithilfe beim Erstellen der Daos und Models.

Diverse Tests und Bugfixes.

Verwendete Technologien

Das Projekt ist in eine dreischichtige Architektur aufgeteilt: Application Layer, Business Layer und Persistence Layer. Im Application Layer wurde JSP verwendet und mit Nutzung von Google's Material Design (<https://material.io/design/>) eine nutzerfreundliche Optik geschaffen. Der Business Layer ist in Java geschrieben und ermöglicht eine Anbindung an den Persistence Layer über eine REST-Schnittstelle. Im Hintergrund arbeitet hier eine MySQL-Instanz, die die Daten dauerhaft speichert.

Die Videofunktionalität wurde mit der Bibliothek VideoJS (<https://videojs.com/>) implementiert. Dieser Player unterstützt eine Vielzahl an Formaten und Einstellungsmöglichkeiten sowie eine gute Anpassbarkeit.

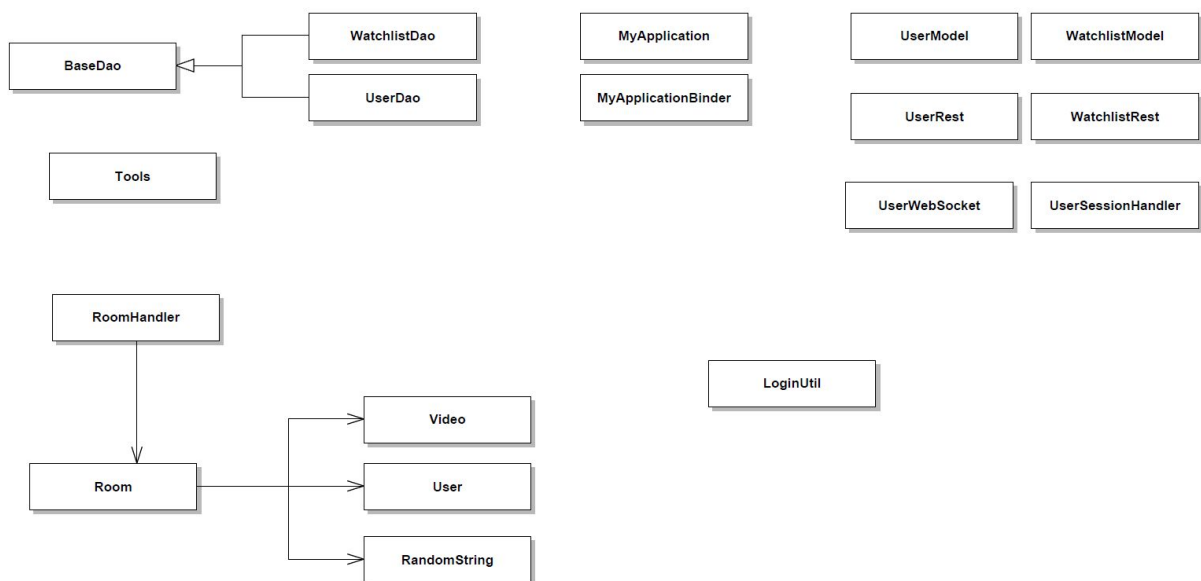
Die Synchronisierungsfunktion wurde mit Hilfe eines Websockets implementiert. Dieser kommuniziert über die Clients mit dem Server, um Statusupdates über den Video-Raum zu sammeln und gegebenenfalls Aktionen durchzuführen. Gibt der Room-Owner beispielsweise den Befehl das Video bei allen Clients des Raums zu starten, wird eine WebSocket-Nachricht an den Server übermittelt. Der Server prüft, ob alle Clients bereit für das starten des Videos sind (bspw. ob genug gebuffered wurde) und gibt dann die Play-Aktion für die Clients frei. Der WebSocket kümmert sich ebenfalls um die Userliste und Playlist des Raumes. In der untenstehenden Abbildung ist beispielhaft die Kommunikation mit dem WebSocket dargestellt.

Data	L...
<code>t{"action":"create","uid":"PXnfh","name":"User 118554","anonymous":false}</code>	72
<code>l{"action":"roomID","id":"D0e06570Z6"}</code>	37
<code>tundefined</code>	9
<code>l{"action":"room-list","userList":[{"uid":"PXnfh","name":"User 118554","avatar":"htt...</code>	2...
<code>t{"action":"video","url":"https://www.youtube.com/watch?v=bYro4P0ORs0"}</code>	70
<code>l{"action":"video","url":"https://www.youtube.com/watch?v=bYro4P0ORs0","curren...</code>	82
<code>l{"action":"playlist","playlist":[{"title":""}]}</code>	47
<code>t{"action":"current","current":0}</code>	32
<code>t{"action":"current","current":1396.0060000419617}</code>	49
<code>t{"action":"play"}</code>	17
<code>t{"action":"current","current":1396}</code>	35
<code>l{"action":"bufferedRequest"}</code>	28
<code>t{"action":"bufferedIndication","readyState":1}</code>	46
<code>t{"action":"current","current":1396}</code>	35

Der Client erstellt über "create" einen Raum und gibt sich als Room-Owner an. Vom der Server verarbeitet die Request und weist dem Raum eine ID zu, die er zurückschickt. Im Anschluss wird die room-list übergeben. Diese enthält Informationen über die User und Playlist des Raumes.

Der Room-Owner schickt anschließend eine Video-Url an den Server, die dieser validiert und an alle Clients zurückschickt. Der Room-Owner schickt einen neuen Startzeitpunkt und anschließend ein Play-Event. Jeder Client schickt dem Server Informationen über den aktuellen Buffer-Status, damit dieser entscheiden kann, ob das Video abgespielt werden darf oder nicht.

Klassendiagramm



Klassendiagramm

In oben stehender Abbildung ist das Klassendiagramm des Projekts zu erkennen. Angehängt ist ein PDF, das das gesamte Diagramm mit Attributen, Methoden und Konstruktoren zeigt.

Einrichtung der Datenbank

Damit das Projekt einwandfrei funktioniert muss vorher die Datenbank eingerichtet werden. Dafür müssen folgende Programme installiert sein, bzw. Pakete heruntergeladen sein: MySQL / Workbench, Wildfly 12, JDBC-Connector.

Zuerst muss in der MySQL Workbench eine Datenbank eingerichtet werden. Nachdem dies passiert ist muss im Wildfly Ordner der Unterordner "bin" geöffnet werden. In diesem befindet sich eine "jboss-cli.bat" Datei. Durch öffnen dieser erscheint eine Eingabeaufforderung. Wichtig ist, dass der Wildfly Server bereits am laufen ist. In dieser wird der Befehl "connect" ausgeführt. Nach der Erfolgsmeldung kann folgender Befehl ausgeführt werden:

```
/subsystem=datasources/jdbc-driver=mysql/:add( \
driver-module-name=com.mysql, \
driver-name=mysql, \
driver-class-name=com.mysql.jdbc.Driver, \
xa-datasource-class=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
```

Wenn dieser Erfolgreich ausgeführt wurde kann folgender ausgeführt werden:

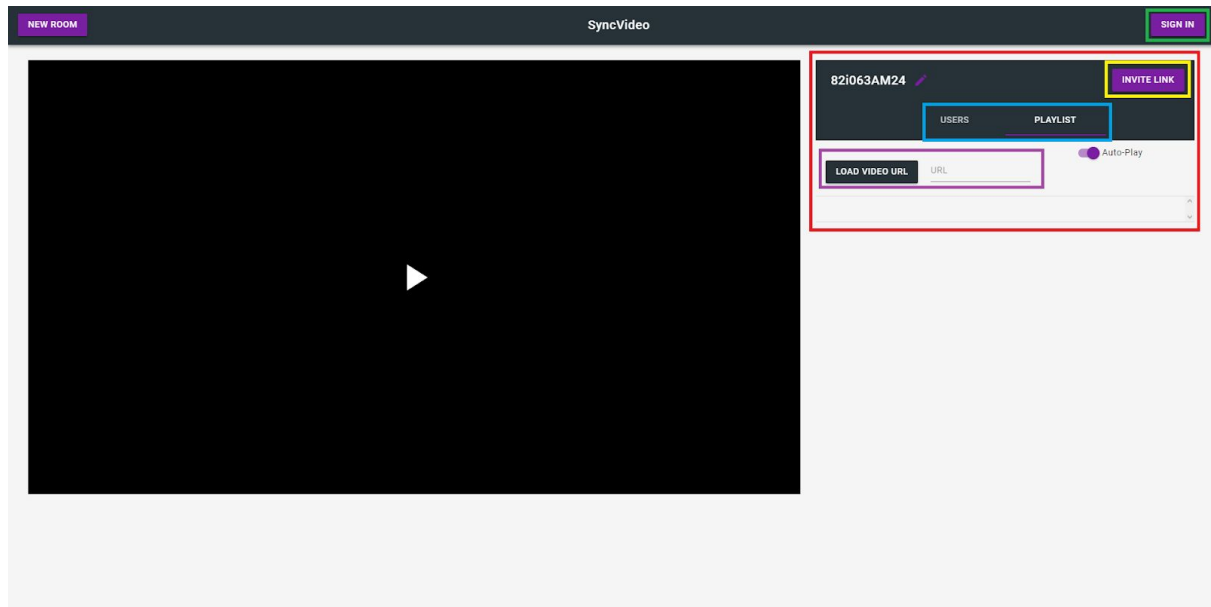
```
/subsystem=datasources/data-source=syncvideoPU/:add( \  
jndi-name=java:jboss/datasources/syncvideoPU, \  
connection-url=jdbc:mysql://localhost:3306/syncvideo, \  
driver-name=mysql, \  
driver-class=com.mysql.jdbc.Driver, \  
enabled=true, \  
jta=true, \  
use-java-context=true, \  
user-name="root", \  
password="pw", \  
)
```

Hierbei ist wichtig, dass in der ersten und zweiten Zeile der Name der Datasource, mit dem Namen in der Persistence.xml des Projektes übereinstimmt. In der dritten Zeile muss die URL der Datenbank richtig angegeben sein. Zuletzt muss in den letzten beiden Zeilen der Username und das Passwort für die Datenbank eingegeben werden.

Nachdem dieser Befehl ausgeführt wurde, sollte der Wildfly Server noch einmal neu gestartet werden, um die Einstellungen zu übernehmen.

User Interface und Bedienung

Falls Wildfly, MySQL und die Entwicklungsumgebung richtig eingerichtet sind und der Server gestartet wurde, so lässt sich das Projekt mit der URL <http://localhost:8080/SyncVideo/> aufrufen. Das User Interface sieht wie folgt aus.

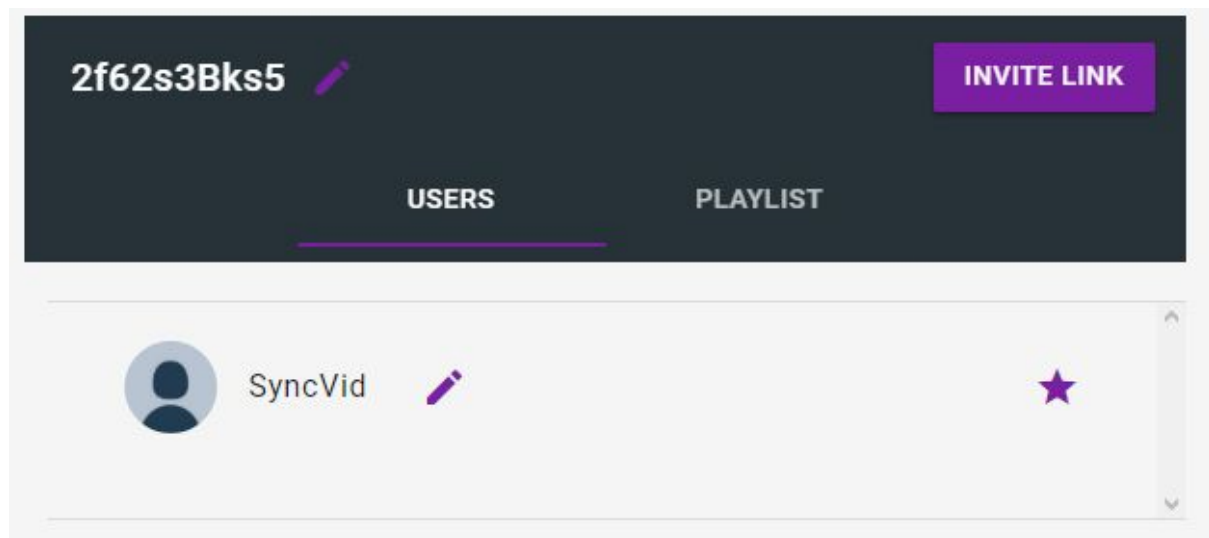


Das User Interface

Die farbigen Kästchen umranden einige wichtige Elemente. Das wichtigste Element ist der linksbündige Video Player (hier noch in schwarz, da noch kein Video eingefügt wurde). In diesem wird später das Video abgespielt. Es handelt sich hierbei um einen frei verfügbaren Video Player.

Der rote Kasten umrahmt das rechtsbündige Element in welchem verschiedene Einstellungen zum aktuellen Raum vorgenommen werden können. Der String oben links im roten Kasten ist der Name des Raums, dieser kann mittels klick auf den Stift rechts daneben beliebig geändert werden. Dadurch entsteht dort ein Texteingabefeld, in welchem der Name geändert werden kann.

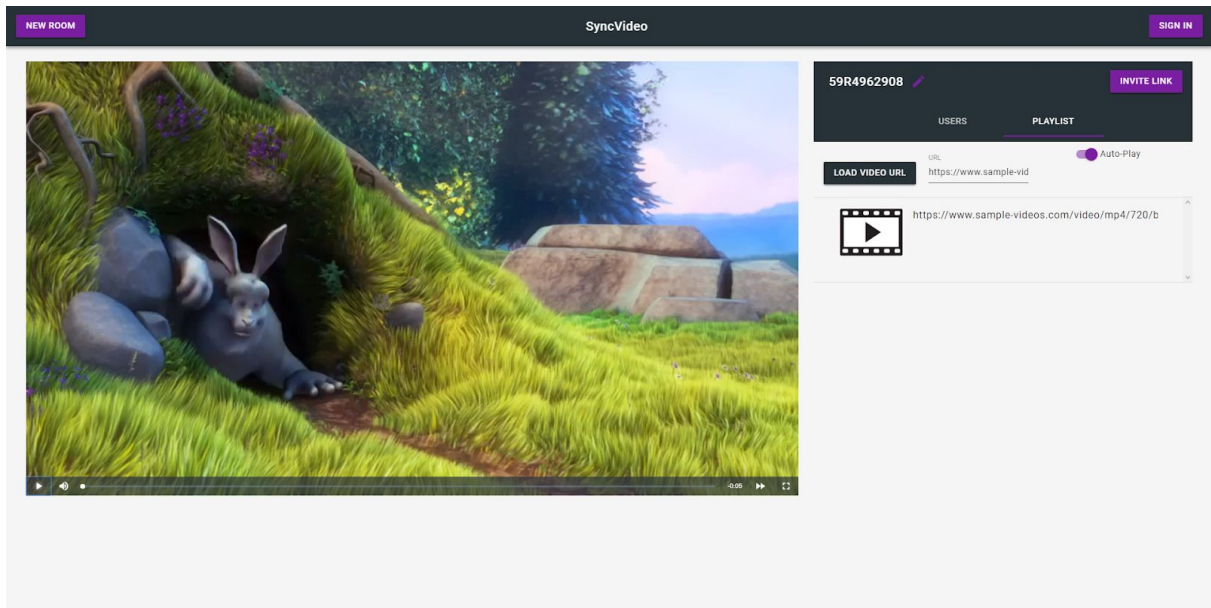
Durch klicken auf eines der beiden Elemente im blauen Kasten lässt sich die Ansicht umschalten. Unter Playlist kann man Videos zur Playlist hinzufügen und unter User kann man alle User sehen, welche sich aktuell im Raum befinden. Das dortige Element sieht wie folgt aus.



Userliste anstatt den Playlstoptionen

Durch klicken auf den Button, welcher von dem gelben Kasten umrandet ist, lässt sich der Link zu dem aktuellen Raum direkt in die Zwischenablage kopieren. Dieser Link kann dann an einen Freund geschickt werden. Wenn dieser diesen öffnet betritt er automatisch den gleichen Raum.

In dem vom Violetten Kasten umrahmten Element lässt sich der Link zu einem Video einfügen, welches synchron angeschaut werden soll. Nach dem Klick auf "Load Video URL" wird das Video geladen und es kann mittels Klick auf den Video Player das Video bei allen Usern gleichzeitig gestartet werden.

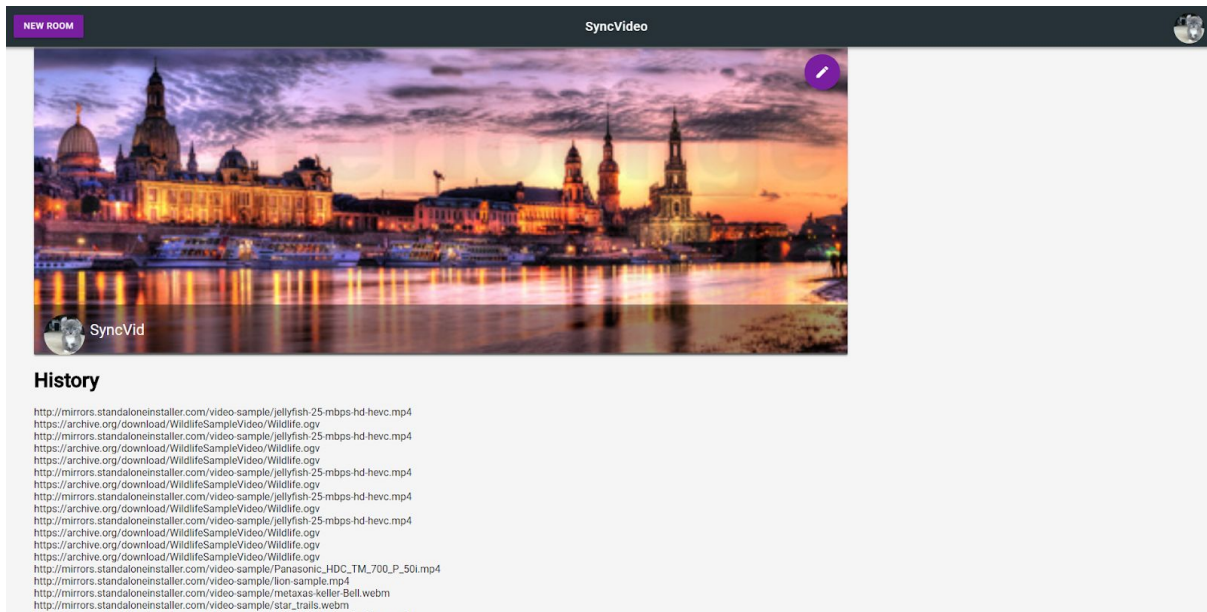


Das Video wurde erfolgreich geladen

Beim Klick auf “Sign In” im grünen Kasten öffnet sich die Login Maske. In dieser kann man sich mit seinem Usernamen und dem entsprechenden Passwort anmelden.

Die Login Maske

Nach dem Einloggen kann man auf dem Bild, welches nun anstelle des Sign In Buttons da ist ein Menü öffnen. In diesem kommt man durch Klicken auf “Profil” auf das eigene Profil.



Die Profilseite mit Avatar und Profilbanner

Auf der Profilseite wird ein Profilbild, ein Banner und die History des Benutzer angezeigt. Der Avatar lässt sich ändern, indem man auf den aktuellen Avatar klickt und in dem Pop-Up-Eingabefeld eine URL zu einem neuen Avatar einfügt und dies bestätigt. Gleiches ist mit dem Banner nach dem Klick auf den Stift oben rechts im Banner möglich. Die History zeigt die zuletzt gesehenen Videos des Benutzers in chronologischer Reihenfolge.