Phornthep Sachdev(5680836) & Arun Bhusri(5680607)
ICCS486
Professional Practices on Network & Technology I
Smartflix CMD application

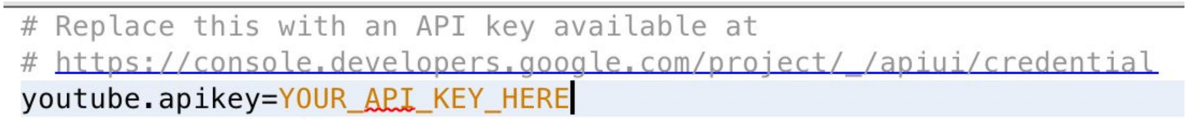
README

Youtube API Authentication

1. You need a Google Account to access the Google Developers Console, request an API key, and register your application.
2. Create a project in the Google Developers Console and obtain authorization credentials so your application can submit API requests.
3. After creating your project, make sure the YouTube Data API is one of the services that your application is registered to use:
    1. Go to the Developers Console and select the project that you just registered.
    2. Open the API Library in the Google Developers Console. If prompted, select a project or create a new one. In the list of APIs, make sure the status is **ON** for the **YouTube Data API v3**.
4. If your application will use any API methods that require user authorization, read the authentication guide to learn how to implement OAuth 2.0 authorization.
5. Select `JAVA` client library to simplify your API implementation.

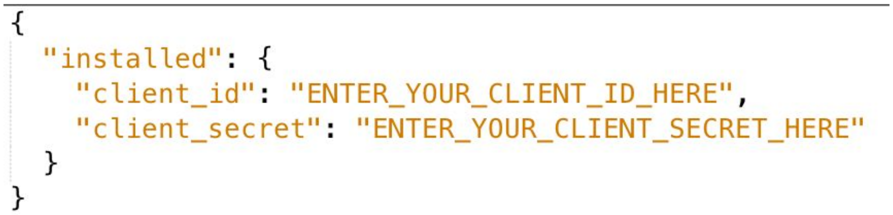
After obtaining both API key and OAuth 2.0 client ID from the Google Cloud Platform, go to resource folder and you see both the files.

In src/main/resources/youtube.properties add:



```
# Replace this with an API key available at
# https://console.developers.google.com/project/_/apiui/credential
youtube.apikey=YOUR_API_KEY_HERE
```

In src/main/resources/client_secrets.json add:



```
{
  "installed": {
    "client_id": "ENTER_YOUR_CLIENT_ID_HERE",
    "client_secret": "ENTER_YOUR_CLIENT_SECRET_HERE"
  }
}
```

Google Cloud Speech API Authentication

Set up service account with the Cloud Speech API:
https://developers.google.com/identity/protocols/application-default-credentials

After Cloud SDK and APC is retrieved:
https://cloud.google.com/speech/docs/auth#application_default_credentials
To export credentials to some location on the desktop.

AWS DYNAMODB:

Table name: favoriteVideos

Sort Key: **username (String)**

Range Key: **video_id (String)**

Provisioned Read Capacity : 1, Provisioned Write Capacity : 1, No Auto-scale

AWS Profile Credentials should have DynamoDBFullAccess Policy.

AWS MySQL RDS:

Database schema is in dbschema.sql

In /src/main/resources/dbConnection.properties, change this part:

```
dbName =YOUR_AWS_RDS_DB_NAME
username =YOUR_AWS_RDS_DB_USERNAME
password =YOUR_AWS_RDS_DB_PASSWORD
hostname =YOUR_AWS_RDS_ENDPOINT
```

Make sure the DB name is DevRDS because in src/main.java/service/SQLQuery.java, function getConnection(), the String jdbcURL ends with `/DevRDS`. It should match the DB name in the dbConnection.properties

BUILDING AND RUNNING:

- Maven must be installed - preferably version 3.3.9

Building: In the project directory run `mvn clean` then `mvn package`

A target folder will be created.

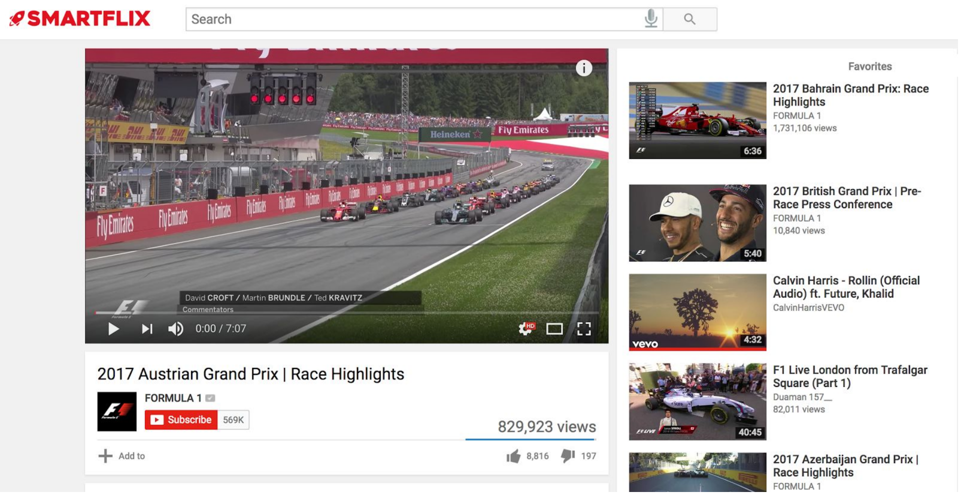
Running: In the target folder run this command `java -jar smartflix-jar-with-dependencies.jar`. This is a fat jar meaning it already comes packed with all dependencies.

PLAN

- Search for videos with different criterias (view count, date, relevance, etc)
- Synchronous voice streaming and real-time conversion
- Be able to save favorite videos
- Play and pause the video
- Skip to different section of the video

A web application where you could do these steps using voice commands. We planned to use HTML/CSS/JS for out front-end, JAVA for back-end and AWS LEX for Chatbot.

The design of the application was to look something like this:


SMARTFLIX
Search
David CROFT / Martin BRUNDLE / Ted KRAVITZ
Commentators
0:00 / 7:07
2017 Austrian Grand Prix | Race Highlights
FORMULA 1
Subscribe
569K
829,923 views
Add to
8,816
197
Favorites
2017 Bahrain Grand Prix: Race Highlights
FORMULA 1
1,731,106 views
6:36
2017 British Grand Prix | Pre-Race Press Conference
FORMULA 1
10,840 views
5:40
Calvin Harris - Rollin (Official Audio) ft. Future, Khalid
CalvinHarrisVEVO
vevo
4:32
F1 Live London from Trafalgar Square (Part 1)
Duaman 157__
82,011 views
40:45
2017 Azerbaijan Grand Prix | Race Highlights
FORMULA 1

OUTCOME

Goals achieved:

- Search for videos with different criteria(view count, rating, relevance, etc)
- Save favorite videos

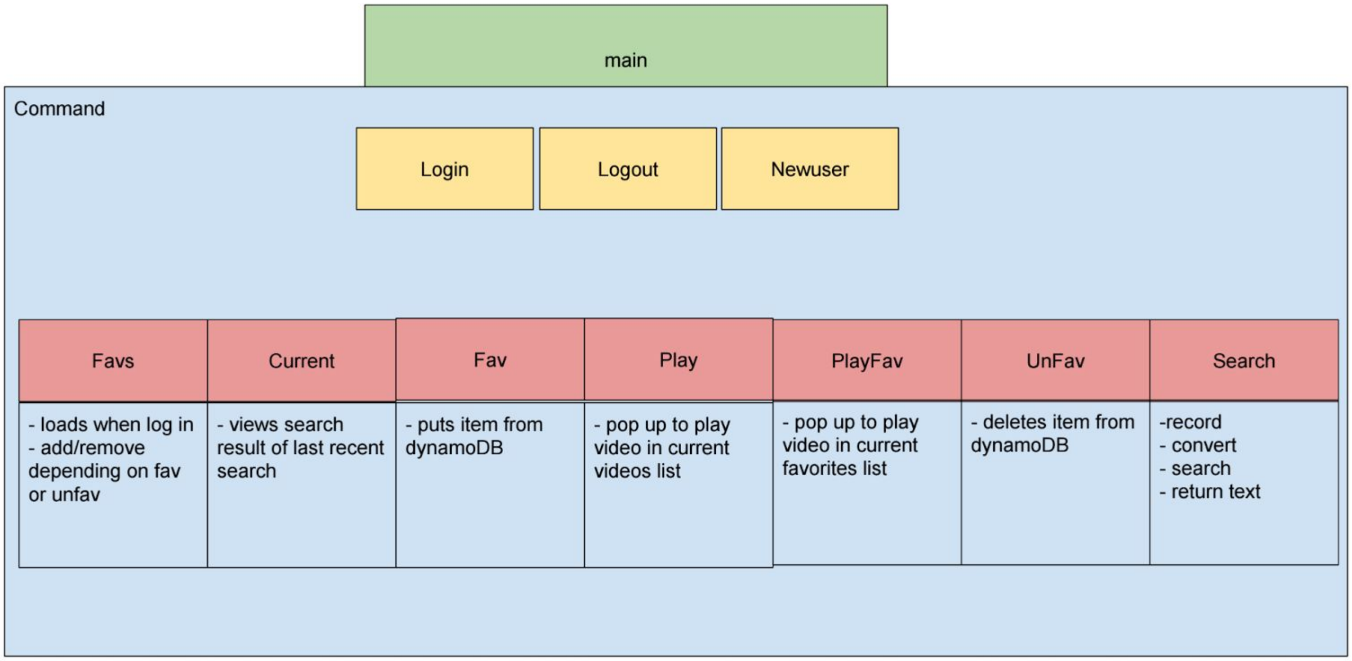
Failed to achieve:

- Synchronous voice streaming and real-time conversion
- Play and pause video using voice commands.
- Skipping to different section of video using voice commands.

Alternative:

- Since we couldn't get the real-time voice conversion, we came up with a way by recording the voice and store it in a .wav file, then send .wav file to back-end to convert it to text. We then use this text to search for video.
- Instead of having JFrame for this application, we just pop up the video in the browser that's on the user's machine.

Architecture



Favs	Current	Fav	Play	PlayFav	UnFav	Search
- loads when log in - add/remove depending on fav or unfav	- views search result of last recent search	- puts item from dynamoDB	- pop up to play video in current videos list	- pop up to play video in current favorites list	- deletes item from dynamoDB	-record - convert - search - return text

Commands

- ➔ **login, newuser, logout**
- ➔ **search <seconds>**
- ➔ **play <playid>**
- ➔ **playfav <favid>**
- ➔ **fav <favid>**
- ➔ **unfav <favid>**
- ➔ **favs**
- ➔ **current <search/videos>**
- ➔ **quit**

Audio file

The Java Sound Recorder function createPath() is called in the record function before recording to make sure the audio directory exists or else it creates it.

Semantics of current video lists

When a search has been successful, the current video list is updated with the number of videos specified from the search. The videos in the list are listed with play id from 0 to the size -1 of the array. We can use the `play <playid>` to play the video. This list does not persist. If a new search is made, it gets replaced with new search results.

Semantics of favorite video lists

A stack is used for this. When the user logs in, the favorite videos of the user is loaded into the stack. When fav <playid> is run the stack is appended. When unfav <favid> is run, the element is removed from the location.

We can use playfav <favid> to have a browser popup of the youtube video.

END

Cannot submit compiled jar as it would not run with the placeholder credentials/properties in src/main/resources.