Shelley Wang CS249 – Project 2

Youtube's Got Talent: A musical mashup Web Service Documentation Youtube

The Youtube API documentation is found here: https://developers.google.com/youtube/

Registering my app

I started by registering my app and obtaining the authentication codes necessary to use this API. Because Youtube is part of Google, this part was similar to registering Fusion Table access for the mobile apps in project one, so this step was relatively easy. (A detailed explanation of the steps can be found here: https://developers.google.com/youtube/registering_an_application)

Code samples

This is an interesting reference with examples of how to use Google APIs (in Javascript): https://developers.google.com/api-client-library/javascript/samples/samples
Specifically, here are sample code snippets for the Youtube API: https://developers.google.com/youtube/v3/code_samples/javascript

You can import their code library to simply using their API. We may do this in the future.

Data API for searching Youtube videos

Right now, we are most interested in the Data API, which allows us to search Youtube videos and find the videos that we are interested in. I am assuming that from the Last.fm API, we are able to retrieve the name of an artist and their song. Using that piece of information, I want to be able to search Youtube videos for covers of that song.

The documentation for using the search portion of the API is found here: https://developers.google.com/youtube/v3/docs/search

There are many different parameters I can set, but right now I am most interested in part (whether you are searching the title of the video or its description), maxResults (number of videos retrieved), and q (query being searched). In the future, we may want to look at parameters to sort the results in order of popularity or use other parameters to narrow down the results further.

The example of a request is this:

The search results are displayed in JSON format.

This is an example of one resulting video that is retrieved:

```
{
"kind": "youtube#searchResult",
"etag": "\"luEKunDQKPL9wdQIy7vaPTsm7fY/fSdNNH7aqvrl3a7by4YrfMEJvrM\"",
"id": {
"kind": "youtube#video",
"videoId": "ALph_u2iee8"
}
}
```

From this, we are able to get the videoId, which is used to then display the video.

Displaying videos

Another part of the Youtube API we are interested in is getting information about a video and displaying it. We would like to be able to embed videos to our site, so you do not have to go to the Youtube site directly to view videos.

The documentation for the Youtube Javascript player is found here: https://developers.google.com/youtube/js_api_reference

One important thing to note is that the Youtube player requires a certain version of Flash, so we need a warning on the page that says that videos will not load unless you have Flash installed.

We use an URL similar to the following to add an embedded Youtube player to our page:

```
http://www.youtube.com/v/VIDEO_ID?version=3&enablejsapi=1
```

The VIDEO ID should be the videoId that we got earlier using the search function.

There are more functions to change the state of the player, for example from paused to playing, but I do not believe we will be using this functionality because we want users to be able to choose what videos they want to start or stop.

Finding information about a video

Here is the information on finding data about a particular video: https://developers.google.com/youtube/v3/docs/videos

There are many different parameters that you can specify for what information you want to be found, such as title, snippet, viewCount, etc... We will need to decide exactly what information we need for our page.

Here's an example of a request that can be sent:

```
GET
https://www.googleapis.com/youtube/v3/videos?id=ALph_u2iee8&part=id%2C+snipp
et%2C+status&key=AIzaSyCcN4ey0somnnylhMsHQD62DeZPOFtneGU
```

Here's the response in JSON format:

```
"id": "ALph u2iee8",
"kind": "youtube#video",
"etag": "\"luEKunDQKPL9wdQIy7vaPTsm7fY/tjYZh7aLgf2ZIOiRZggb8YKg1zU\"",
"snippet": {
"publishedAt": "2011-10-09T12:20:54.000Z",
"channelId": "UCgc00bfF_PvO_2AvqJZHXFg",
"title": "Journey - Faithfully (Boyce Avenue acoustic cover) on iTunes",
"description": "iTunes: http://iTunes.com/BoyceAvenue\nTickets: http://BoyceAvenue.com/Tour\nMerchandise:
http://BoyceAvenue.com\n\Tickets:\n4/12 Snow College http://bit.ly/c7Ah1q\n4/13 SLC at In the Venue
http://bit.ly/YDVrtK\n\nBoyce Avenue cover of Journey's song \"Faithfully\"\n\nAudio Produced by Alejandro
Manzano & Daniel Manzano\nArrangement by Alejandro\nRecorded & Engineered by Adam Barber\nMixed &
Mastered by Adam Barber\nAdditional Mixing & Mastering by Alejandro & Daniel\nVideo Produced by Daniel &
Alejandro\n\nFilmed at Clear Track Studios\nhttp://www.cleartrackstudios.com
\n\nFacebook\nhttp://facebook.com/boyceavenue\n\nTwitter\nhttp://twitter.com/boyceavenue\n\nYouTube\nhttp://y
outube.com/BoyceAvenue (Music Channel)\nhttp://youtube.com/BoyceAvenueExtras (Vlog Channel)\n\nOfficial
Site\nhttp://boyceavenue.com",
"thumbnails": {
"default": {"url": "https://i.ytimg.com/vi/ALph_u2iee8/default.jpg"},
"medium": {"url": "https://i.ytimg.com/vi/ALph_u2iee8/mqdefault.jpg"},
"high": {"url": "https://i.ytimg.com/vi/ALph_u2iee8/hqdefault.jpg"}
"channelTitle": "boyceavenue",
"categoryId": "10"
},
"status": {
"uploadStatus": "processed",
"privacyStatus": "public",
"embeddable": true,
"publicStatsViewable": false,
"license": "youtube"
```