Anirudha Joshi – SEC01 (NUID 002991365)

Big Data System Engineering with Scala
Spring 2023
Assignment (Working with API)

GitHub - https://github.com/anirudhajoshi2808/Anirudha_Joshi_CSYE7200

- Questions:

You may use Scala requests to get the Spotify Playlist data in JSON format.

The ID of the playlist is : 5Rrf7mqN8uus2AaQQQNdc1

The Spotify developer API is : https://developer.spotify.com/console/get-playlist/Links to an external site.

Playlist : https://open.spotify.com/playlist/5Rrf7mqN8uus2AaQQQNdc1Links to an external site.

In the JSON response, you will have the duration_ms field. This is the duration of each song in milliseconds. You are to find the top 10 longest songs. Each song will have an Artist name and an artist ID. There may be multiple artists for the same song. You are to query the artist details of every artist in the top 10 longest songs and display them in order of their followers.

The Spotify API to query the artist is : https://developer.spotify.com/console/get-artist/Links to an external site.

You must call this API for every artist ID in the top 10 longest songs.

The expected output would be :

Part 1)

Songname1 , duration_ms

Songname2 , duration_ms

……

Songname10 , duration_ms

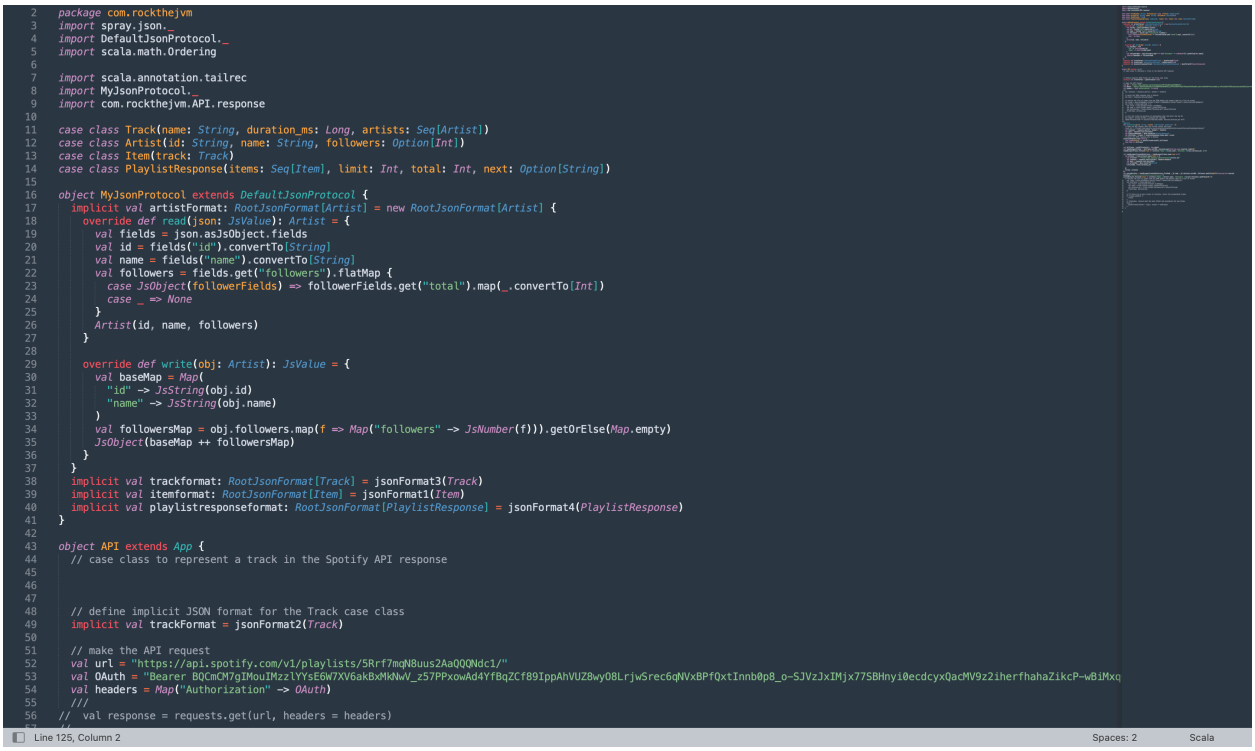
Part 2)

Artist1 : follower_count

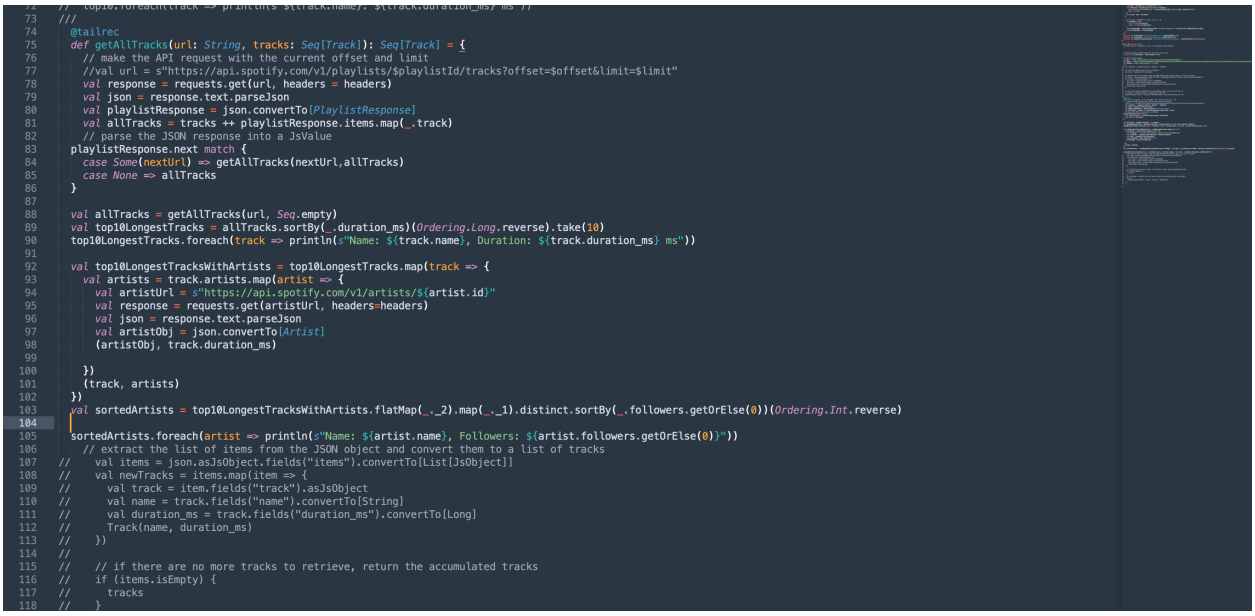
Artist2 : follower_count … etc

- Code

API has a Limit of 100 songs, below code overrides that limit of 100 and scans all 500 songs to get the result



```scala
package com.rockthejvm
import spray.json._
import DefaultJsonProtocol._
import scala.math.Ordering

import scala.annotation.tailrec
import MyJsonProtocol._
import com.rockthejvm.API.response

case class Track(name: String, duration_ms: Long, artists: Seq[Artist])
case class Artist(id: String, name: String, followers: Option[Int])
case class Item(track: Track)
case class PlaylistResponse(items: Seq[Item], limit: Int, total: Int, next: Option[String])

object MyJsonProtocol extends DefaultJsonProtocol {
  implicit val artistFormat: RootJsonFormat[Artist] = new RootJsonFormat[Artist] {
    override def read(json: JsValue): Artist = {
      val fields = json.asJsObject.fields
      val id = fields("id").convertTo[String]
      val name = fields("name").convertTo[String]
      val followers = fields.get("followers").flatMap {
        case JsObject(followerFields) => followerFields.get("total").map(_.convertTo[Int])
        case _ => None
      }
      Artist(id, name, followers)
    }

    override def write(obj: Artist): JsValue = {
      val baseMap = Map(
        "id" -> JsString(obj.id)
        "name" -> JsString(obj.name)
      )
      val followersMap = obj.followers.map(f => Map("followers" -> JsNumber(f))).getOrElse(Map.empty)
      JsObject(baseMap ++ followersMap)
    }
  }
  implicit val trackformat: RootJsonFormat[Track] = jsonFormat3(Track)
  implicit val itemformat: RootJsonFormat[Item] = jsonFormat1(Item)
  implicit val playlistresponseformat: RootJsonFormat[PlaylistResponse] = jsonFormat4(PlaylistResponse)
}

object API extends App {
  // case class to represent a track in the Spotify API response



  // define implicit JSON format for the Track case class
  implicit val trackFormat = jsonFormat2(Track)

  // make the API request
  val url = "https://api.spotify.com/v1/playlists/5Rrf7mqN8uus2AaQQQNdc1/"
  val OAuth = "Bearer BQCmCM7gIMouIMzzlYYsE6W7XV6akBxMkNwV_z57PPxowAd4YfBqZCf89IppAhVUZ8wyO8LrjwSrec6qNVxBPfQxtInnb0p8_o-SJVzJxIMjx77SBHnyi0ecdcyxQacMV9z2iherfhahaZikcP-wBiMxq
  val headers = Map("Authorization" -> OAuth)
  ///
//  val response = requests.get(url, headers = headers)
```



```
// top10.foreach(track => println(s"${track.name}, ${track.duration_ms} ms"))
///
  @tailrec
  def getAllTracks(url: String, tracks: Seq[Track]): Seq[Track] = {
    // make the API request with the current offset and limit
    //val url = s"https://api.spotify.com/v1/playlists/$playlistId/tracks?offset=$offset&limit=$limit"
    val response = requests.get(url, headers = headers)
    val json = response.text.parseJson
    val playlistResponse = json.convertTo[PlaylistResponse]
    val allTracks = tracks ++ playlistResponse.items.map(_.track)
    // parse the JSON response into a JsValue
  playlistResponse.next match {
    case Some(nextUrl) => getAllTracks(nextUrl,allTracks)
    case None => allTracks
  }

  val allTracks = getAllTracks(url, Seq.empty)
  val top10LongestTracks = allTracks.sortBy(_.duration_ms)(Ordering.Long.reverse).take(10)
  top10LongestTracks.foreach(track => println(s"Name: ${track.name}, Duration: ${track.duration_ms} ms"))

  val top10LongestTracksWithArtists = top10LongestTracks.map(track => {
    val artists = track.artists.map(artist => {
      val artistUrl = s"https://api.spotify.com/v1/artists/${artist.id}"
      val response = requests.get(artistUrl, headers=headers)
      val json = response.text.parseJson
      val artistObj = json.convertTo[Artist]
      (artistObj, track.duration_ms)

    })
    (track, artists)
  })
  val sortedArtists = top10LongestTracksWithArtists.flatMap(_._2).map(_._1).distinct.sortBy(_.followers.getOrElse(0))(Ordering.Int.reverse)

  sortedArtists.foreach(artist => println(s"Name: ${artist.name}, Followers: ${artist.followers.getOrElse(0)}"))
    // extract the list of items from the JSON object and convert them to a list of tracks
//    val items = json.asJsObject.fields("items").convertTo[List[JsObject]]
//    val newTracks = items.map(item => {
//      val track = item.fields("track").asJsObject
//      val name = track.fields("name").convertTo[String]
//      val duration_ms = track.fields("duration_ms").convertTo[Long]
//      Track(name, duration_ms)
//    })
//
//    // if there are no more tracks to retrieve, return the accumulated tracks
//    if (items.isEmpty) {
//      tracks
//    }
```

SBT:

```
name := "scala-at-light-speed"

version := "0.1"
libraryDependencies += "com.lihaoyi" %% "requests" % "0.8.0"
libraryDependencies += "io.spray" %% "spray-json" % "1.3.6"

scalaVersion := "2.13.1"
```

- Answer:

Top 10 longest songs Name:

- Name: The End - New Stereo Mix Advanced Resolution, Duration: 699533 ms
- Name: Desolation Row, Duration: 681400 ms
- Name: Free Bird, Duration: 550066 ms

- Name: Purple Rain, Duration: 520786 ms
- Name: Kashmir - 1990 Remaster, Duration: 508200 ms
- Name: Iron Man - Live, Duration: 500693 ms
- Name: Good Times, Duration: 495400 ms
- Name: Stairway to Heaven - 1990 Remaster, Duration: 478173 ms
- Name: L.A. Woman, Duration: 471160 ms
- Name: A Whiter Shade Of Pale - Live, Duration: 466506 ms

Artists of top 10 songs in order of their followers Name:

- Name: Led Zeppelin, Followers: 13148831
- Name: The Doors, Followers: 7105007
- Name: Black Sabbath, Followers: 6806788
- Name: Prince, Followers: 6644900
- Name: Bob Dylan, Followers: 6019697
- Name: Lynyrd Skynyrd, Followers: 4517304
- Name: CHIC, Followers: 645895
- Name: Procol Harum, Followers: 400400
- Name: The Danish National Concert Orchestra and Choir, Followers: 75