

Hrishikesh Sanjay Pawar – SEC01 (NUID 002707307)

# Big Data System Engineering with Scala

## Spring 2023

### Assignment: Working with APIs



### **-List of Tasks Implemented**

Used Scala requests to get the Spotify playlist in JSON format and found the top 10 longest songs. Also queried the artist details of every song and displayed the artist name in order of their followers.

### **-Code**

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help CSYE7200 [...]Big Data\CSYE7200-Hrishikesh-Pawar] - SpotifyRequest.scala [assignment-working-with-api]
CSYE7200-Hrishikesh-Pawar > assignment-working-with-api > src > main > scala > SpotifyRequest.scala
build.sbt x SpotifyRequest.scala
1 import spray.json._
2 import scala.annotation.tailrec
3
4 case class Artist(id: String, name: String, followers: Option[Int])
5
6 case class Song(name: String, duration_ms: Long, artists: Seq[Artist])
7
8 case class Item(track: Song)
9
10 case class PlaylistResponse(items: Seq[Item], limit: Int, total: Int, next: Option[String])
11
12 object MyJsonProtocol extends DefaultJsonProtocol {
13   implicit val artistFormat: RootJsonFormat[Artist] = new RootJsonFormat[Artist] {
14     override def write(obj: Artist): JsValue = {
15       val baseMap = Map(
16         "id" -> JsString(obj.id),
17         "name" -> JsString(obj.name)
18       )
19       val followersMap = obj.followers.map(f => Map("followers" -> JsNumber(f))).getOrElse(Map.empty)
20       JsObject(baseMap ++ followersMap)
21     }
22   }
23 }
Run: SpotifyApi x
Pushed 1 commit to origin/Spring2023 (4 minutes ago)
5°C Rain 11:1 CRLF UTF-8 2 spaces Spring2023 01:58 14-03-2023
```

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help CSYE7200 [...]Big Data\CSYE7200-Hrishikesh-Pawar] - SpotifyRequest.scala [assignment-working-with-api]
CSYE7200-Hrishikesh-Pawar > assignment-working-with-api > src > main > scala > SpotifyRequest.scala
build.sbt x SpotifyRequest.scala
22
23 def read(json: JsValue): Artist = {
24   val fields = json.asJsObject.fields
25   val id = fields("id").convertTo[String]
26   val name = fields("name").convertTo[String]
27   val followers = fields.get("followers").flatMap {
28     case JsObject(followersFields) => followersFields.get("total").map(_.convertTo[Int])
29     case _ => None
30   }
31   Artist(id, name, followers)
32 }
33
34 implicit val trackFormat: RootJsonFormat[Song] = jsonFormat3(Song)
35 implicit val itemFormat: RootJsonFormat[Item] = jsonFormat1(Item)
36 implicit val playlistResponseFormat: RootJsonFormat[PlaylistResponse] = jsonFormat4(PlaylistResponse)
37
38
39 object SpotifyApi extends App {
40   val token = "BQAU2BfbvNMp0oCzcXs_Imp0YqY65o0jdsd-QnKXcRPHntKRdvKtr6t4jk7HzA11k0BmrVNVucpysQYM3azRxfjU30tdktv4RSj7uzHt26DSWWHd0cDz817E7iKKAB636o"
41   val url = "https://api.spotify.com/v1/playlists/5Rrf7mqN8uvs2AaQQNdc1/tracks?offset=0&limit=100"
42   val headers = Map(
43     "Authorization" -> s"Bearer $token"
44   )
45 }
Run: SpotifyApi x
Pushed 1 commit to origin/Spring2023 (4 minutes ago)
5°C Rain 11:1 CRLF UTF-8 2 spaces Spring2023 01:58 14-03-2023
```

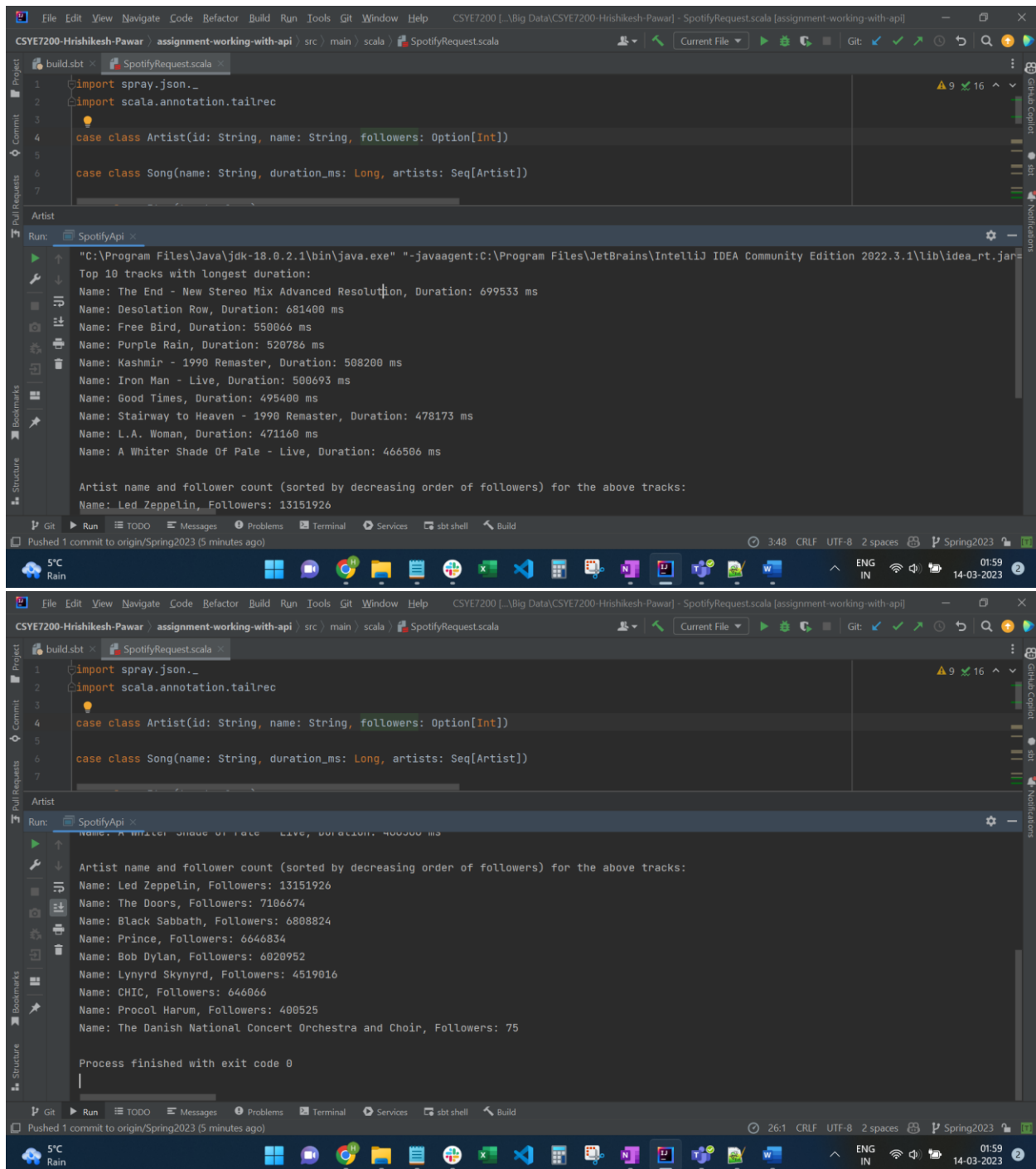
```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help CSYE7200 [...] SpotifyRequest.scala [assignment-working-with-api]
CSYE7200-Hrishikesh-Pawar | assignment-working-with-api | src | main | scala | SpotifyRequest.scala
build.sbt SpotifyRequest.scala
37 }
38
39 object SpotifyApi extends App {
40   val token = "BQAU2BfbvNMp0oCzcxS_Imp0YqY65o0jdsd-QnKXcRPHntKRdVktR6t4jk7HzA1k0BmrVNVucpysQYM3azRxfjU30tdktv4RSj7UzHt260SWWhDcDz817E7iKKAB636o"
41   val url = "https://api.spotify.com/v1/playlists/5Rrtf7mqN8vvs2AaQQQNdc1/tracks?offset=0&limit=100"
42   val headers = Map(
43     "Authorization" -> s"Bearer $token"
44   )
45
46   import MyJsonProtocol._
47
48   @tailrec
49   def getAllTracks(url: String, tracks: Seq[Song]): Seq[Song] = {
50     val response = requests.get(url, headers = headers)
51     val json = response.text.parseJson
52     val playlistResponse = json.convertTo[PlaylistResponse]
53     val allTracks = tracks ++ playlistResponse.items.map(_.track)
54
55     if (playlistResponse.next.isDefined) {
56       getAllTracks(playlistResponse.next.get, allTracks)
57     } else {
58       allTracks
59     }
60   }
61
62   val allTracks = getAllTracks(url, Seq.empty)
63   val top10LongestTracks = allTracks.sortWith(_.duration_ms > _.duration_ms).take(10)
64
65   println("Top 10 tracks with longest duration: ")
66   top10LongestTracks.foreach(track => println(s"Name: ${track.name}, Duration: ${track.duration_ms} ms"))
67
68   val top10LongestTracksWithArtists = top10LongestTracks.map(track => {
69     val artists = track.artists.map(artist => {
70       val artistUrl = s"https://api.spotify.com/v1/artists/${artist.id}"
71       val response = requests.get(artistUrl, headers = headers)
72       val json = response.text.parseJson
73       val artistObj = json.convertTo[Artist]
74       (artistObj, track.duration_ms)
75     })
76   })
77 }
```

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help CSYE7200 [...] SpotifyRequest.scala [assignment-working-with-api]
CSYE7200-Hrishikesh-Pawar | assignment-working-with-api | src | main | scala | SpotifyRequest.scala
build.sbt SpotifyRequest.scala
55 if (playlistResponse.next.isDefined) {
56   getAllTracks(playlistResponse.next.get, allTracks)
57 } else {
58   allTracks
59 }
60 }
61
62 val allTracks = getAllTracks(url, Seq.empty)
63 val top10LongestTracks = allTracks.sortWith(_.duration_ms > _.duration_ms).take(10)
64
65 println("Top 10 tracks with longest duration: ")
66 top10LongestTracks.foreach(track => println(s"Name: ${track.name}, Duration: ${track.duration_ms} ms"))
67
68 val top10LongestTracksWithArtists = top10LongestTracks.map(track => {
69   val artists = track.artists.map(artist => {
70     val artistUrl = s"https://api.spotify.com/v1/artists/${artist.id}"
71     val response = requests.get(artistUrl, headers = headers)
72     val json = response.text.parseJson
73     val artistObj = json.convertTo[Artist]
74     (artistObj, track.duration_ms)
75   })
76 })
77 }
```

The screenshot shows an IDE window with the following details:

- File Explorer:** Project structure includes `src` (main, scala) and `SpotifyRequest.scala`.
- Code Editor:** Contains Scala code for `object SpotifyApi extends App`.
  - Lines 71-79: Fetches artist data from `https://api.spotify.com/v1/artists/{artist.id}`, parses the JSON, and converts it to an `Artist` object.
  - Line 81: Commented-out line: `// val sortedArtists = top10LongestTracksWithArtists.flatMap(_._2).map(_._1).distinct.sortBy(_._followers.getOrElse(0))(Ordering.Int.reverse)`
  - Lines 82-87: Processes `top10LongestTracksWithArtists` by flattening, mapping, and sorting by follower count in descending order.
  - Lines 89-90: Prints the artist name and follower count for the top 10 tracks.
- Run Console:** Shows the command `Run: SpotifyApi`.
- Status Bar:** Indicates 91:1 CRLF UTF-8 2 spaces, and the commit message "Pushed 1 commit to origin/Spring2023 (5 minutes ago)".
- System Tray:** Shows weather (5°C Rain) and date/time (01:59 14-03-2023).

## -Unit tests/Output



## - GitHub Link

<https://github.com/hrishikesh-pawar/CSYE7200-Hrishikesh-Pawar/tree/Spring2023/assignment-working-with-api>

