

Use Spotify Audio Features To Categorize Music

Practical Exam



Goal

Spotify provides an API for basic track information:

- **ID**, e.g. *2IEcSduKEXEK5KJ9hJzICz*
- **name**, e.g. *Gloana Bauer (Teenage Dirtbag)*
- **artist**, e.g. *D' Hundskrippln*
- ...

as well as related audio features:

- **energy**, e.g. *0.454*
Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.
- **speechiness**, e.g. *0.0388*
Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
- **loudness**, e.g. *-8.758*
The overall loudness of a track in decibels (dB). Loudness values are averaged.
- **tempo**, e.g. *97.532*
The overall estimated tempo of a track in beats per minute (BPM)
- **acousticness**, e.g. *0.268*
A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
- ...

<https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features>



Goal

We want to make use of those audio features to automatically assign each track to a certain category:

Metal
Electro

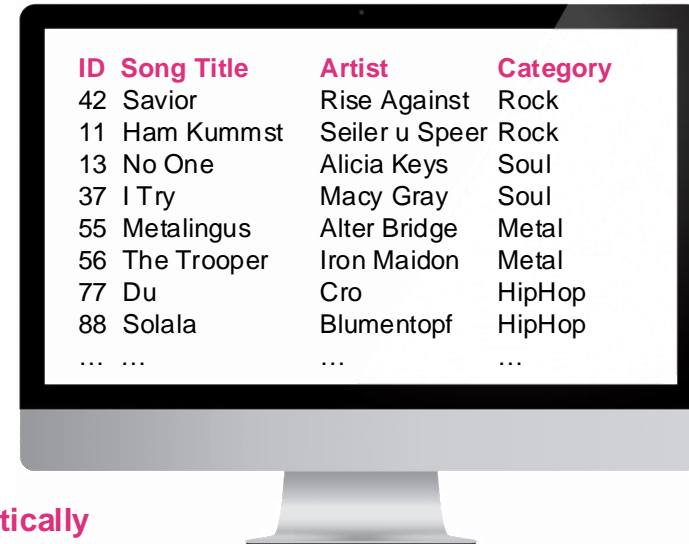
Classic
Podcast

Rock
Soul

Vocal
HipHop

Workflow:

- **Query** data from Spotify API
- **Save** raw data (JSON files) to HDFS
- **Optimize, reduce** and **clean** raw data and save it to **final** directory on HDFS
- **Calculate** categories (*Metal, Classic, Rock, ...*)
- **Join** track information and audio features and **save** everything to **end-user database** (e.g. MySQL, MongoDB...)
- Provide a simple HTML Frontend which reads from end-user database and displays result
- The whole data workflow **must be implemented** within an ETL **workflow tool** (e.g. Pentaho Data Integration or Airflow) and **run automatically**



| ID | Song Title | Artist | Category |
|-----|-------------|----------------|----------|
| 42 | Savior | Rise Against | Rock |
| 11 | Ham Kummst | Seiler u Speer | Rock |
| 13 | No One | Alicia Keys | Soul |
| 37 | I Try | Macy Gray | Soul |
| 55 | Metalingus | Alter Bridge | Metal |
| 56 | The Trooper | Iron Maidon | Metal |
| 77 | Du | Cro | HipHop |
| 88 | Solala | Blumentopf | HipHop |
| ... | ... | ... | ... |

Dataflow: 1. Get Track Information

```
curl -X "GET" "https://api.spotify.com/v1/tracks/2IEcSduKEXEK5KJ9hJzlCz?market=DE" -H "Accept: application/json" -H "Content-Type: application/json" -H "Authorization: Bearer BBB0v7vCOHXxaUAshFUVIFbozLDO_ysq8cPb4wYR3oko_JfDcrUSEsy0Mq6P4cu5vvS0ljm6R24raME8o4qa2XNy02lhGGCufMgwGptf43s2OoAcbfJUfcsXA1-dpW19_x_3rG75ADnA4dlr25"
```

1



<https://api.spotify.com/v1/tracks/{id}>

<https://api.spotify.com/v1/audio-features/{id}>

2

```
[...]  
"album":{  
  "album_type":"single",  
  "artists":[  
    {  
      "href":"https://api.spotify.com/v1/artists/4x4nSliAvh7RM7dVxbs9aP",  
      "id":"4x4nSliAvh7RM7dVxbs9aP",  
      "name":"D'HundskrippIn",  
      "type":"artist",  
      "uri":"spotify:artist:4x4nSliAvh7RM7dVxbs9aP"  
    }  
  ]  
}  
"name":"Gloana Bauer (Teenage Dirtbag)",  
"release_date":"2016-07-01"  
[...]  
"href":"https://api.spotify.com/v1/tracks/2IEcSduKEXEK5KJ9hJzlCz", "id":"2IEcSduKEXEK5KJ9hJzlCz",  
[...]
```

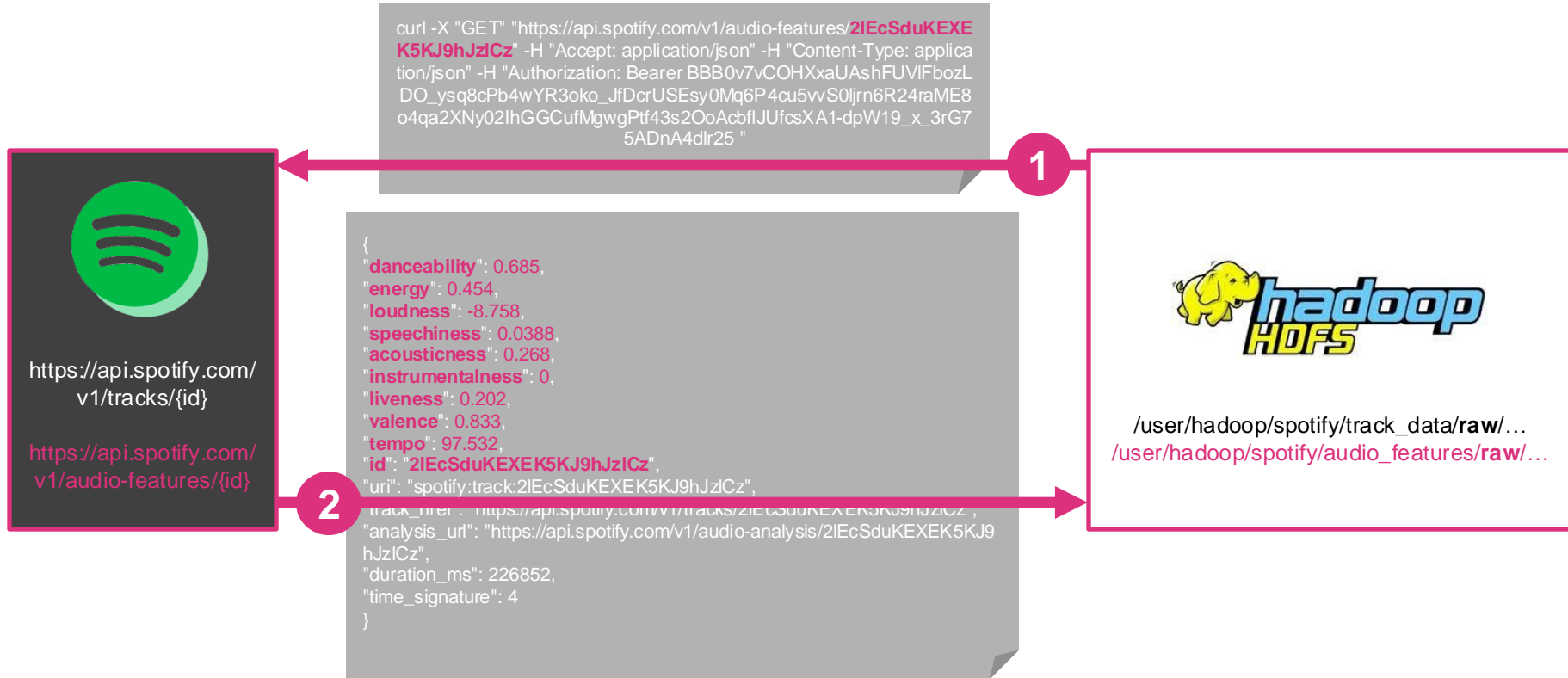


/user/hadoop/spotify/track_data/raw/...
/user/hadoop/spotify/audio_features/raw/...

<https://developer.spotify.com/documentation/web-api/reference/#/operations/get-track>



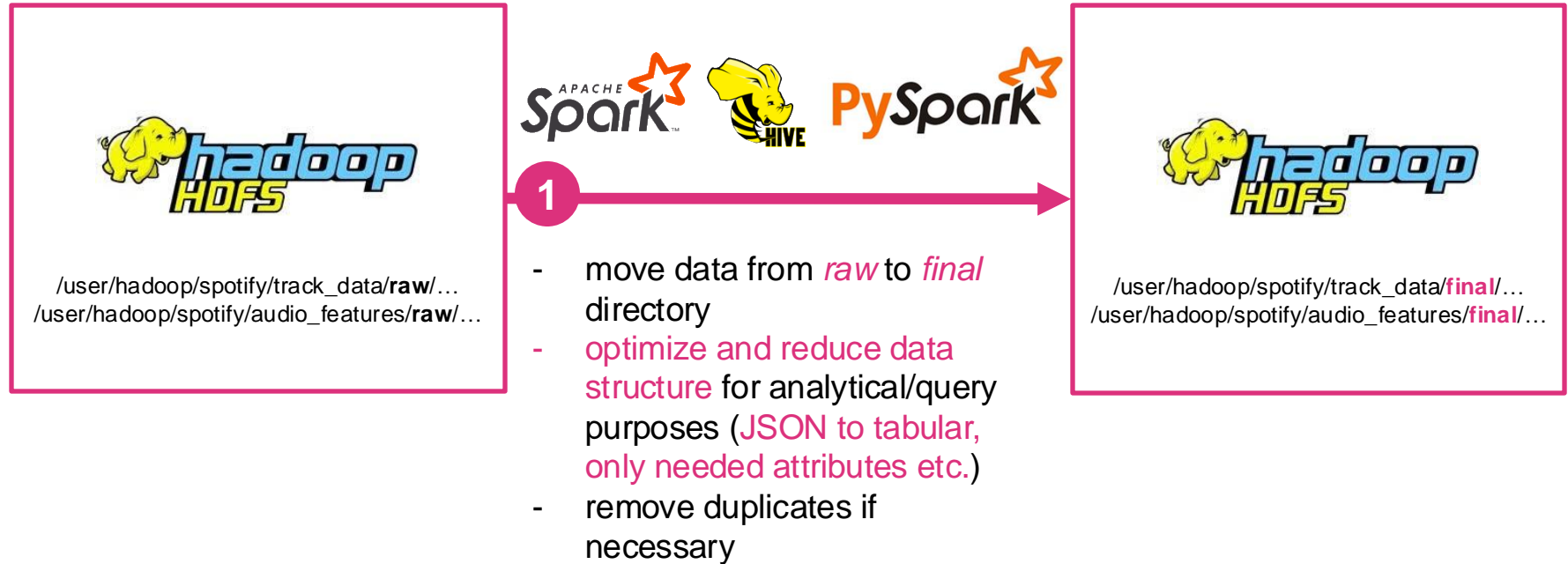
Dataflow: 2. Get Track Audio Features



<https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features>



Dataflow: 3. Raw To Final Transfer



Dataflow: 4. Run Analysis and Save Results



/user/hadoop/spotify/track_data/final/...
/user/hadoop/spotify/audio_features/final/...



1

- calculate categories for each track, using *Hive*, *Python*, *Spark* or *PySpark*
- join track and audio feature data
- save everything to a end-user database (e.g. *MySQL*, *mongoDB*)



Dataflow: 5. Provide Simple Web Interface



1

- Create a simple Website which reads and displays data from end-user database

| ID | Song Title | Artist | Category |
|-----|-------------|----------------|----------|
| 42 | Savior | Rise Against | Rock |
| 11 | Ham Kummst | Seiler u Speer | Rock |
| 13 | No One | Alicia Keys | Soul |
| 37 | I Try | Macy Gray | Soul |
| 55 | Metalingus | Alter Bridge | Metal |
| 56 | The Trooper | Iron Maidon | Metal |
| 77 | Du | Cro | HipHop |
| 88 | Solala | Blumentopf | HipHop |
| ... | ... | ... | ... |