



SPOTIFY TO SQLITE

Desarrollo de Aplicaciones para la Ciencia de Datos

Segundo Curso

Escuela de Ingeniería Informática y Matemáticas

Universidad de Las Palmas de Gran Canaria

ADONAI OJEDA MARTÍN

9/11/2022

VERSIÓN 1

ÍNDICE:

1. Resumen…………………………………………………………………….3
2. Recursos utilizados…………………………………………………….4
3. Diseño…………………………………………………………………………5
4. Conclusiones………………………………………………………………6
5. Líneas Futuras……………………………………………………………7
6. Bibliografía…………………………………………………………………8

RESUMEN

Spotify contiene una api-rest accesible para los usuarios y nos hemos basado en hacer operaciones rest para trabajar con ella.

Acceder a los datos y a la información de los cantantes a través de java y el api de Spotify, ha sido nuestro objetivo en esta práctica.

A continuación, se ha cogido estos datos y se han representado en una base de datos con distintas tablas usando el lenguaje JDBC y DML para poder analizar.

Finalmente, podemos ver estos datos estructurados en las tablas de donde podemos comparar los artistas y sacar conclusiones con respecto a ellos

RECURSOS UTILIZADOS

El trabajo se ha realizado en el entorno de desarrollo Intellij IDEA CE, con la versión java 11 jdk.

Y para su desarrollo y obtener todas las versiones futuras y pasadas del proyecto se ha subido a un repositorio en GitHub.

Java Doc 11 ha sido la herramienta de documentación usada.

DISEÑO

Model View Controller es un estilo de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

Este estilo es el que usamos para estructurar el proyecto, dividimos el paquete “Model” con las clases POJO, “Artist”, “Track” y “Album”. En el paquete “View” quedaría implementar una visión gráfica del proyecto en un futuro, pero de momento queda vació. Por otro lado, el paquete “Controller”, cuyas clases son el “Controller” y el “Main”, y finalmente, las clases “SpotifyAccessor” donde hacemos peticiones a la api y “SpotifyAuthorization” que permite hacer las peticiones

CONCLUSIONES

En conclusión, este trabajo nos ha permitido familiarizarnos con el api rest de Spotify y poder mejorar nuestros conceptos sobre las operaciones o peticiones que este tipo de api nos permite.

Mientras que la implementación nos ha puesto problemas de excepciones y errores nuevos que tratar.

Finalmente, no podemos dejar atrás alguna sorpresa sobre la popularidad de los cantantes que nos ha impactado.

Líneas Futuras

Nuestra api podría ser de gran utilidad en el mercado comercial, pero no sin implementar alguna que otra mejora como podría ser una vista gráfica para poder representar todos estos datos de una manera más intuitiva.

Por otro lado, habría que mirar alguna manera para poder tratar mayor volumen de datos y que el tiempo de respuesta sea equitativo.

BIBLIOGRAFÍA

<https://developer.spotify.com/documentation/web-api/>

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

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

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

<https://www.sqlitetutorial.net/sqlite-java/>