Spotify a SQLite

2º de GCID

Desarrollo de Aplicaciones para Ciencia de Datos

Escuela de Ingeniería Informática

ULPGC • UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Autor del proyecto

Jia Hao Yang

Colaboradores

Ninguno

Versión actual del proyecto: 1.0
Versión actual de la memoria: 1.0
Revisiones totales a día 07/11/2002 de la memoria: 1

Resumen

Para realizar el proyecto, se podía hacer de dos maneras. La primera de ella era utilizando los métodos POJO que se vieron en las clases o la segunda utilizando un JsonParser que como su nombre indica, lo que hace es parsear el Json en una String de manera que podamos acceder a sus respectivos valores en función de la clave dada.

En mi caso he optado por la segunda opción, no he utilizado ninguna clase POJO que represente lo que devuelve la API pero sí he utilizado un JsonParser para conseguir toda la información que me parecía más conveniente para más adelante introducirlo en una base de datos no relacional.

El código funcionara de la siguiente manera:

- En la clase Main, este solo llamara a la clase Controller que realiza lo siguiente.

- Controller tiene un Hashmap con los respectivos artistas y un método llamado execute que llama a los 3 métodos principales de las clases Artist, Album y Track para el funcionamiento del programa.

- En Artist tenemos un método que se llama insertIntoArtistTable que su función principal es insertar datos en la base de datos. Dentro tiene otro método llamado getArtist que toma el ID de los artistas de la clase Controller. Dentro del método getArtist, este contiene otro método que se llama addToArtistLists que simplemente añade a listas los diferentes valores de cada atributo. Y finalmente un método llamado updateAll que actualiza la base de datos por columnas

- El resto clases (Album y Track) realizan prácticamente la misma función que la clase Artist.

- Existe otra clase que se llama DatabaseManager que realiza todo lo que tenga que ver con la base de datos. Este a su vez extiende a otra clase llamada Updater que realiza todos los updates necesarios en la base de datos para rellenarlo en función del formato que tengan.

Índice

Recursos utilizados

Entornos de desarrollo

El entorno de desarrollo utilizado ha sido IntelliJ IDEA Edu en su última versión (2022.2.2) desarrollada por JetBrains.

También se han utilizado otras herramientas de soporte para el desarrollo como Visual Studio Code.

Herramientas de control de versiones

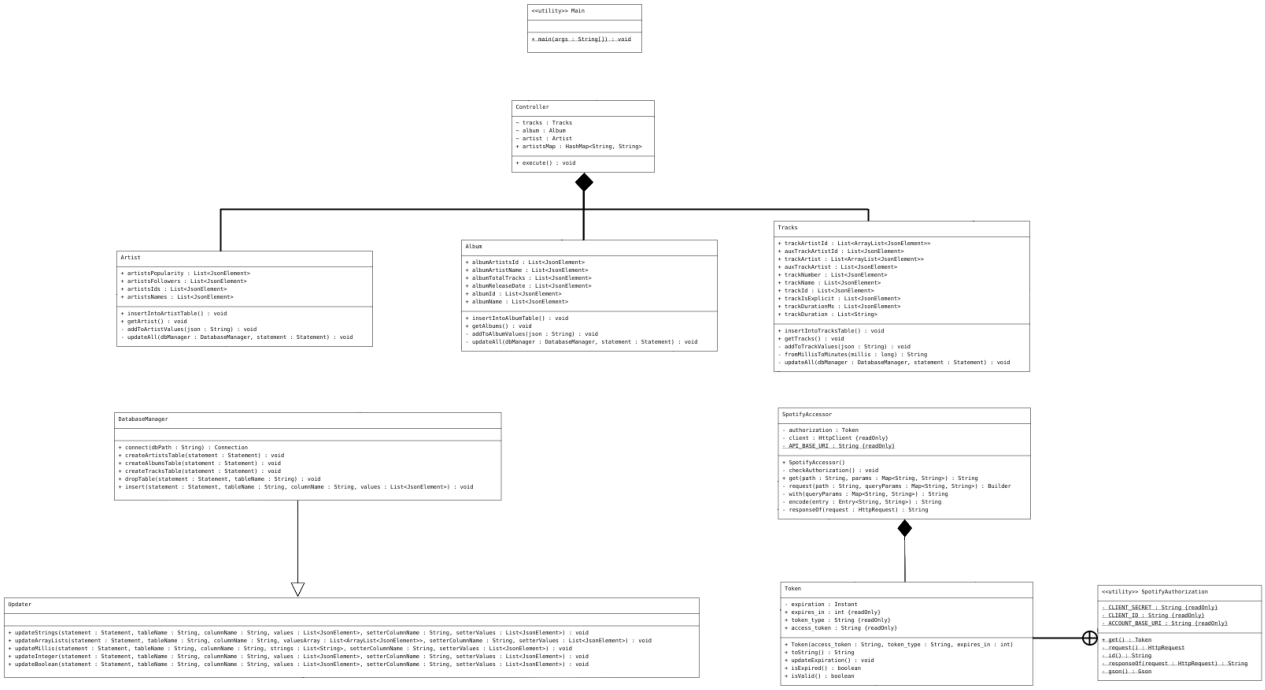
Para el control de versiones se utiliza la herramienta Git desarrollada por Linus Torvalds.

Herramientas de documentación

Para la documentación de la memoria, se utilizó LibreOffice Writter.

Diseño

Diagrama de clases



Conclusiones

Lecciones aprendidas y experiencias

Durante el desarrollo de este proyecto, he aprendido múltiples lecciones y conceptos nuevos que no conocía previamente. Algunas de las lecciones que se han aprendido pueden ser los siguientes:

1. Convertir objetos Json pasándolo por diferentes tipos para obtener un resultado deseado.
2. Parsear objetos Json para poder manipularlo con mayor facilidad.
3. Conectarse a una base de datos, crearla, insertarle datos y actualizarla para rellenarlo con los datos que nosotros queramos.
4. Obtener los datos a través de la API de Spotify con el *CLIENT_ID* y el *SECRET_CLIENT_ID*.

Además, se ha obtenido más experiencias en el uso y manejo en el entorno de desarrollo IntelliJ. manejo y detección de excepciones, búsqueda de información desconocida, aplicar el manejo de bases de datos como SQLite etc...

Que hacer si tuviera que volver a empezar

Para empezar, cambiaría todo el código para rellenar la base de datos SQLite por filas en lugar de por columnas que es como esta hecho el programa ya que rellenarlo por columnas es altamente mas ineficiente que por filas. Debido a la falta de tiempo y de que en el ultimo día de la entrega me he dado cuenta de la alta in eficiencia del código, no he sido capaz de implementar el Prepared Statement para la inserción de datos, así que, ese seria uno de los cambios principales del programa.

Líneas futuras

Para versiones posteriores de este proyecto, podemos implementar por ejemplo una base de datos que sea relacional, optimizar el código, aumentar la eficiencia en la inserción de información en la base de datos investigando qué técnicas puede ser mejor, estructurar mejor el código, implementar un motor gráfico si en un futuro se llega a comercializar, alojarlo en un servidor para tenerlo en la nube etc.

Bibliografía

Las referencias que se han tomado para la realización del proyecto son:

[1] https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_class_System_JsonParser.htm

[2] https://www.javatpoint.com/java-string-format

[3] https://www.javatpoint.com/java-list

[4] https://www.sqlitetutorial.net/

[5] https://inloop.github.io/sqlite-viewer/#

[6] https://open.spotify.com/

[7] https://developer.spotify.com/documentation/web-api/reference/#/operations/get-an-artist

[8] https://www.geeksforgeeks.org/hashmap-put-method-in-java/