



ULPGC

Universidad de
Las Palmas de
Gran Canaria

Escuela de
Ingeniería Informática



Using Spotify API to populate a sqlite database

ULPGC

2º curso

Grado en Ciencia e Ingeniería de Datos

Desarrollo de Aplicaciones para Ciencia de Datos

Jaime Ballesteros Domínguez

09/11/2022

Primera versión

Última revisión: 09/11/2022

Resumen

En este trabajo se nos pedía hacer uso de nuestros conocimientos sobre bases de datos y estructuras de datos POJO con la intención de aplicarlos a la creación de una serie de tablas de datos con información sobre una serie de artistas, sus álbumes y las canciones que estos contienen.

Comencé con la extracción de mi id de usuario y la realización en la clase Main de los diversos métodos de obtención de datos para los distintos apartados siendo el `getArtists` el método para obtener los datos sobre los artistas, `getAlbums` para la adquisición de los contenidos y características de los álbumes de dichos artistas, `getTracks` para extraer las canciones contenidas en los álbumes y `getTracksInfo` para finalmente recoger todos los datos solicitados sobre dichas canciones.

Posteriormente pasé a la creación de la clase Database, en la cual introduje los procesos de diseño de las tablas de las tres clases definidas en Main. Asimismo, añadí los distintos métodos insert con los campos de información que creí convenientes (popularidad del artista requerido, fecha de publicación de determinado álbum, duración de la canción en cuestión) mediante la subinterfaz `PreparedStatement`. Cabe destacar que en las funciones "insert" se especificó que, si se entrase en conflicto en el caso de que dos ids fueran iguales, se copiaría la información que acompaña al nuevo id, en el antiguo.

Seguidamente trasladé los distintos métodos descritos en el segundo párrafo, de la clase Main a la clase `GetFunctions`, siendo esta la encargada de realizar las llamadas pertinentes a Database para que se creen las diferentes tablas de datos se introduzca toda la información en ellas.

Para finalizar, en la clase `Controller` cree una interfaz de texto sencilla la cual posee dos funciones; la función uno permite al usuario introducir los ids de más cantantes cada vez que pulsa dicha tecla y la función dos ejecuta el proceso de creación de las distintas tablas con los datos de los artistas introducidos a través de la interfaz más los cinco que ya están en el código. Cualquier otra tecla introducida fuera de 1 y 2 daría error y la ejecución del programa terminaría.

A la hora de ejecutarlo, la clase Main llamaría a `Controller`, iniciando así la interfaz. En caso de que el usuario quisiera introducir el id de algún cantante más este se guardaría en el `HashMap` "artists". Si quisiera comenzar con la creación de la tabla y el guardado de los datos se realizaría la función `callCenter`, la cual entraría en contacto con `GetFunctions` y esta buscaría la información deseada en la API de Spotify. Finalmente, se llamaría a la clase Database la cual crearía las tablas e insertaría los datos obtenidos de Spotify en estas.

Índice

Resumen.....	3
Recursos utilizados.....	5
Diseño.....	5
Conclusiones	6
Líneas futuras	6
Bibliografía	7

Recursos utilizados

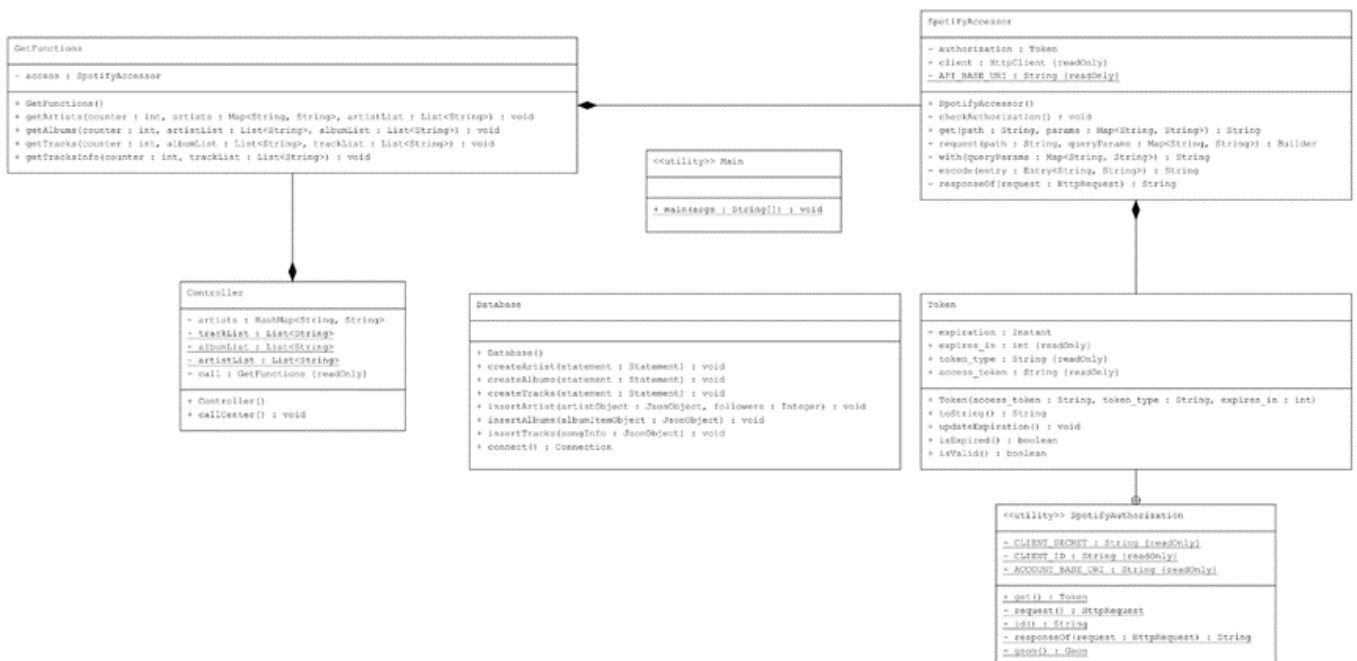
Entornos de desarrollo: El entorno de desarrollo utilizado para este trabajo fue IntelliJ IDEA desarrollado por JetBrains.

Herramientas de control de versiones: El software de control de versiones utilizados ha sido Git.

Herramientas de documentación: Para documentar el trabajo realizado se ha utilizado el software de tratamiento de textos Microsoft Word.

Diseño

En este trabajo se intentó seguir el estilo de diseño Model View Controller (MVC), siendo la clase GetFunctions un ejemplo de Model al manejar los datos de la API de Spotify, Database estaría relacionado con View, al ser la encargada de representar la información recogida de GetFunctions en las tablas de datos y la clase Controller, como su nombre indica, recibiría las órdenes del usuario y se encargaría de estar en contacto con el Model y el View.



Aquí se puede apreciar el diagrama de clases resultante, el cual se puede ver con mayor calidad en el repositorio.

Conclusiones

En el proceso de realización de este trabajo he podido aprender tanto métodos como conceptos nuevos y formas de trabajar con ellos. Algunos ejemplos pueden ser:

- El uso de json, tanto el hecho de transformar datos en elementos, objetos o arrays de json como el parseo de dichos objetos para manejarlos con más facilidad.
- La creación y tratamiento de tablas de datos, siendo la función insert la principal.
- El desarrollo de interfaces de texto sencillas y el manejo de datos obtenidos a través del teclado.
- La descarga de información de una API como es la de Spotify.

Líneas futuras

Para que el producto obtenido se pueda convertir en uno comercializable se tendría que añadir la posibilidad de que el usuario pueda pedir los datos del artista, sus álbumes y las canciones de estos de manera independiente, ya que no todos los clientes van a querer un estudio completo de los cantantes seleccionados además de poder pedirlos con el nombre del artista en vez de con su id. También se tendría que dar la posibilidad en la interfaz de borrar contenido y de eliminar las propias tablas para que el usuario pueda empezar desde cero cuando quiera. Otra mejora podría ser la ampliación de datos que se ofrecen sobre los artistas, como canciones más populares, artistas relacionados.

Seguidamente, la mejora de la visualización de los datos en tabla es otro aspecto para tratar, ya que varias columnas precisan de un espacio que no tienen, tales como los id o los nombres de álbumes y canciones. Además, la interfaz tendría que ser mejorada y ampliada para ofrecer más servicios y para tener una interacción más sencilla.

Por último, intentaría buscar la forma de reducir los tiempos de ejecución ya que, aunque se tenga que descargar mucha información, se tarda bastante en realizar las funciones.

Bibliografía

<https://jenkov.com/tutorials/java-json/gson-jsonparser.html>

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

<https://iteritory.com/read-interactive-command-line-input-with-java-scanner-class/>

<https://www.tabnine.com/code/java/methods/org.json.JSONObject/getJSONArray>

<https://stackoverflow.com/questions/1568762/accessing-members-of-items-in-a-jsonarray-with-java>

<https://www.sqlitetutorial.net/sqlite-create-table/>

<https://www.digitalocean.com/community/tutorials/thread-sleep-java>

https://www.youtube.com/watch?v=_zIfW-4x1ps