



## SPOTIFY API

Using Spotify API to populate a sqlite database

Escuela de Ingeniería Informática - ULPGC

Desarrollo de Aplicaciones para Ciencia de Datos

2º Curso Grado en Ciencia e Ingeniería de Datos

## Resumen

El programa consiste en hacer uso de la API de Spotify con el objetivo de crear una base de datos obteniendo información de esta plataforma.

En primer lugar, se creó una clase Main con un mapa de los artistas con sus nombres e ids para sacar sus canciones, álbumes y características como artistas.

Luego, se creó una interfaz (MusicSource) para crear tres métodos en los que se les pase como parámetro los ids de los artistas y crear tres listas: listas albums, artists y tracks, métodos albumof, trackof, artistof.

Se crearon las clases POJO de Album, Track y Artist en la que se definen sus respectivos atributos.

Posteriormente, se creó una clase (Spotify) que implementa MusicSource, y es la responsable de obtener la información JSON de Spotify y de pasar esta misma al tipo de objetos definido en las clases POJO. Hereda los métodos de la interfaz que implementa y almacena la información en Java Objects en las listas tracks, albums y artists definidas en la interfaz MusicSource. Esta clase será la encargada de devolver las listas mencionadas.

Luego tuvo lugar la creación de una interfaz (MusicDataBase) que es la responsable de crear los métodos encargados de añadir la información a la base de datos (void add(Track track), void add(Album album), void add(Artist artist)).

También, se creó una clase (DMLTranslator), y es encargada de hacer los inserts y diseñar los statements.

Se creó una clase (SQLiteStore) que implementa MusicDataBase, encargada de crear conexión con la base de datos, iniciar esta misma y crear tablas SQLite, además de heredar los métodos de MusicDataBase y en estos crear los statements.

Después se volvió al Main, se crearon tres listas -del mismo nombre que en la interfaz MusicSource- y se hizo un bucle con for anidados, el primer for recorre el mapa y saca los ids, y se los pasa a las funciones creadas en MusicSource: crea una lista de álbumes y otra de canciones de todos los artistas y una lista de estos mismos-luego la clase Spotify se encargará de devolver cada lista en formato Java Objects-.

Dentro de este mismo bucle, se hacen tres bucles paralelos: se encargan de añadir la lista de albums, tracks y artistas ya en Java Objects a la base de datos, con las funciones de la interfaz MusicDataBase. Recorren cada lista, y van elemento a elemento añadiendo. Al usar las funciones de MusicDataBase, se crea comunicación con la clase SQLiteStore para crear la base de datos.

## Índice

Recursos Utilizados.....	3
Entornos de desarrollo .....	3
Herramientas de Documentación .....	3
Diseño .....	3
Conclusiones.....	3
Líneas futuras .....	4
Bibliografía.....	4

## Recursos Utilizados

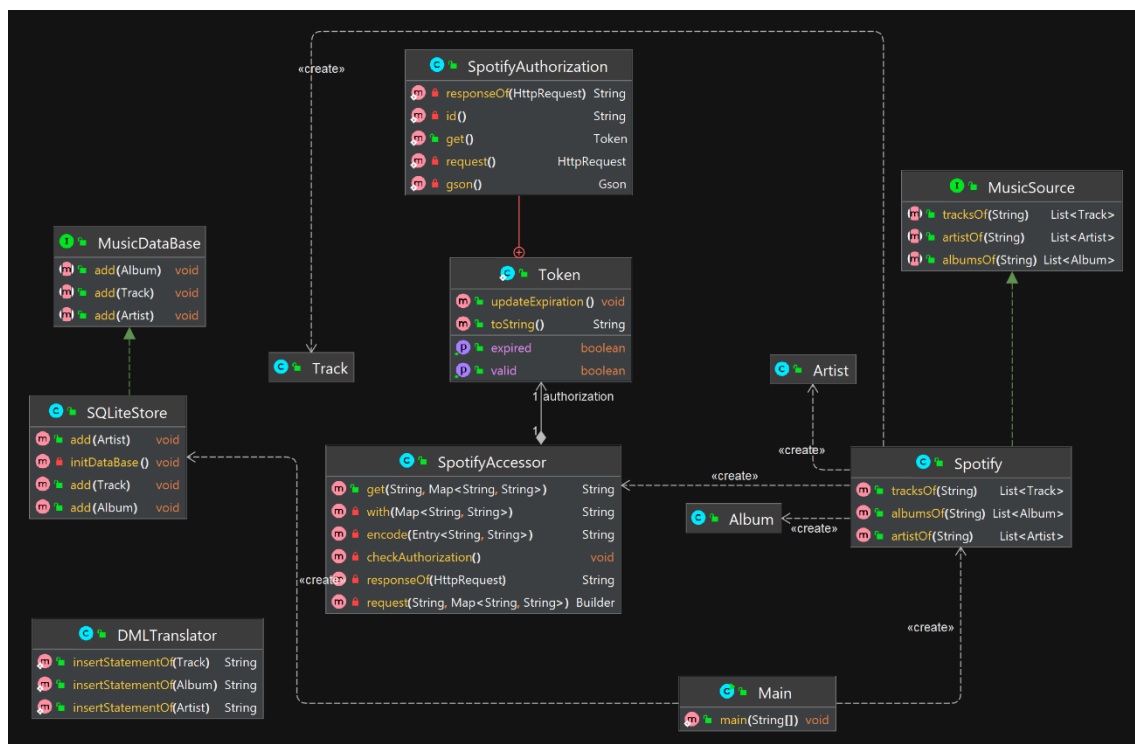
### Entornos de desarrollo

IntelliJ IDEA Community Edition 2022.2.2, IntelliJ IDEA 2022.2.3 (Ultimate).

### Herramientas de Documentación

Microsoft Word 2021.

## Diseño



## Conclusiones

En primer lugar, antes de intentar cualquier obtención de datos del JsonObject a través de la función “get()”, primero revisaría exhaustivamente la estructura del mismo, vería dentro de los atributos que contiene cuáles son concretamente arrays y cuáles son JsonObject, porque al principio cometía el error de usar la función “get()” para un JSONArray cuando esta misma no se puede usar para este tipo de objeto, ya que primero tendría que pasarlo a un JsonObject.

También debería tener en cuenta usar más la depuración, porque es un recurso muy útil que no sabía usar, y me pudo permitir conocer la estructura de los JsonObjects con facilidad, y conocer la causa de los errores, sobre todo los sintácticos.

Por otra parte, debería haber usado PreparedStatement en vez de crear los statements, ya que facilita el trabajo: evita errores, y agiliza la ejecución.

Igualmente, me di cuenta que no eran necesarias las clases POJO, pero no está de más su uso.

Por último, es muy importante recordar que las comillas simples provocan errores a la hora de insertar información a la base de datos y por ello hay que hacer un escape de estas mismas, o eliminarlas.

## Líneas futuras

En primer lugar, es muy importante que el cliente pueda elegir el artista en vez de usar unos predeterminados, y que lo haga mediante el nombre, no el id.

Además, en vez de hacer una tabla de álbumes y otra de canciones, creo que sería más útil hacer una tabla de álbumes por cada artista, e igual con el caso de las canciones.

Asimismo, la duración de las canciones la pasaría a segundos en vez de usar milisegundos; aunque es una medida menos rigurosa, creo que es más adecuada para un usuario ordinario que para un desarrollador.

## Bibliografía

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

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

[https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Conditional Operator](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Conditional_Operator)

[https://www.youtube.com/watch?v=HSuVtkdej8Q&ab\\_channel=BrianFraser](https://www.youtube.com/watch?v=HSuVtkdej8Q&ab_channel=BrianFraser)

[https://www.w3schools.com/java/ref\\_string\\_replace.asp](https://www.w3schools.com/java/ref_string_replace.asp)

<https://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean/>

Miriam Méndez Romero

08/11/2021 1.0