Zápočtový Program

Music Library API

Dávid Buko

Zimný semester, II. Ročník Bc.

Šk. Rok: 2023/2024

Obsah

[1 Úvod 1](#_Toc167148980)

[1.1 Stručné zadanie 1](#_Toc167148981)

[2 Používateľská časť 1](#_Toc167148982)

[2.1 Spustenie 1](#_Toc167148983)

[2.2 Vstupné dáta 2](#_Toc167148984)

[2.3 Výstupné dáta 2](#_Toc167148985)

[2.4 Orientácia v aplikácii 2](#_Toc167148986)

[3 Popis programu 3](#_Toc167148987)

[3.1. Knižnice a moduly 3](#_Toc167148988)

[3.2. Hlavný prúd vykonávania 4](#_Toc167148989)

[3.3. Funkcie správy dát 4](#_Toc167148990)

[3.4. Výstup a užívateľské rozhranie 4](#_Toc167148991)

[3.5. Integrovanie s externým API 5](#_Toc167148992)

[3.6. Spracovanie chýb 5](#_Toc167148993)

[3.7. Bezpečnosť a správa citlivých údajov 5](#_Toc167148994)

[3.8. Rozšíriteľnosť a údržba 5](#_Toc167148995)

[4 Záver 6](#_Toc167148996)

# 1 Úvod

Od malička som bol obklopený hudbou a vždy som ju počúval, keď som mal príležitosť. Moje nadšenie pre hudbu a technológie ma viedlo k vývoju tejto aplikácie, ktorej cieľom je spojiť hudbu s modernými technológiami a poskytnúť jednoduché, no funkčné rozhranie pre správu hudobných záznamov a integráciu s populárnou službou Spotify. Tento projekt bol inšpirovaný mojou vášňou pre hudbu a mojou snahou pochopiť, ako môžem efektívne využívať dostupné technologické nástroje na zlepšenie môjho hudobného zážitku.

## 1.1 Stručné zadanie

Cieľom projektu je poskytnúť používateľom jednoduché rozhranie CLI (command-line interface) na správu osobnej hudobnej knižnice a integráciu so Spotify API. Aplikácia umožňuje používateľom pridávať skladby, vytvárať playlisty, vyhľadávať hudbu a spravovať svoje hudobné súbory prostredníctvom príkazov zadaných v termináli.

# 2 Používateľská časť

## 2.1 Spustenie

Pre spustenie aplikácie je nutné mať nainštalovanú Javu vo verzii 11 alebo vyššej. Aplikáciu je možné spustiť z terminálu alebo príkazového riadku v operačnom systéme, kde užívateľ musí navigovať do priečinka obsahujúceho súbor MusicLibraryAPI.jar a spustiť ho príkazom java -jar MusicLibraryAPI.jar. Pred spustením je dôležité uistiť sa, že všetky závislosti definované v pom.xml sú správne nakonfigurované a dostupné.

Celý projekt vrátane detailných inštrukcií je voľne dostupný v GitHub repozitári: <https://github.com/Chills16/MusicLibraryAPI>. V prípade akýchkoľvek problémov alebo nejasností pri spúšťaní programu prosím pozrite na README.md súbor.

## 2.2 Vstupné dáta

Aplikácia očakáva od užívateľa zadanie textových príkazov prostredníctvom konzoly. Tieto príkazy zahŕňajú napríklad pridanie novej skladby, vyhľadávanie existujúcich skladieb, vytvorenie alebo úprava playlistov. Inštrukcie pre vstupné dáta sú jasne definované textom v termináli, podľa zvolenej voľby užívateľom. Pri spustení programu je voľba vybrať prvú inštrukciu, napísaním čísla 1-11 do terminálu.

## 2.3 Výstupné dáta

Výstupom aplikácie sú textové informácie zobrazené v konzole, ktoré poskytujú spätnú väzbu na príkazy zadané používateľom. Výstupy zahŕňajú potvrdenia o úspešnom pridaní skladieb, vytvorení alebo zmene playlistov, výsledky vyhľadávania, a chybové hlášky, ak akcia nevyjde podľa očakávaní. Pri každom príkaze aplikácia jasne komunikuje, či bola operácia úspešná alebo informuje o dôvode neúspechu.

## 2.4 Orientácia v aplikácii

Orientácia v aplikácii je navrhnutá tak, aby bola čo užívateľsky najpríjemnejšia aj pre tých, ktorí nemajú skúsenosti s CLI aplikáciami. Po spustení aplikácie sú užívateľom zobrazené možnosti, medzi ktorými môže vyberať:

1. Pridať skladbu
2. Zobraziť zoznam skladieb
3. Odstrániť skladbu
4. Vytvoriť playlist
5. Pridať skladbu do playlistu
6. Odstrániť skladbu z playlistu
7. Zobraziť playlist
8. Zmazať playlist
9. Vyhľadať skladby
10. Vyhľadať playlisty
11. Ukončiť aplikáciu

Užívateľ vyberá možnosť zadaním príslušného čísla príkazu do konzoly a potvrdí ho stlačením klávesy Enter. Po zvolení akcie sú mu podľa potreby zobrazené ďalšie inštrukcie, ako napríklad požiadavka na zadanie mena umelca alebo názvu skladby. Po spracovaní príkazu aplikácia vždy zobrazí výsledok alebo ďalšie možnosti, čo uľahčuje navigáciu a používanie aplikácie. Toto rozhranie zaisťuje, že používatelia môžu efektívne spravovať svoju hudobnú zbierku bez potreby rozsiahlych technických znalostí.

# 3 Popis programu

## 3.1. Knižnice a moduly

Aplikácia je rozdelená do niekoľkých modulov a využíva externé knižnice, ktoré zlepšujú jej funkčnosť a zjednodušujú kód:

dataAccessObject (DAO): Tento modul obsahuje rozhranie SongDAO a jeho implementáciu InMemorySongDAO. Tieto triedy sú zodpovedné za správu údajov piesní, ako je ukladanie, odstraňovanie a vyhľadávanie piesní v pamäti.

model: Obsahuje triedy Song a Playlist, ktoré definujú štruktúru dátových objektov používaných v aplikácii.

service: Modul MusicLibraryService poskytuje logiku aplikácie pre správu piesní a playlistov, vrátane interakcie s Spotify API na získavanie informácií o piesňach.

spotify: Obsahuje triedy SpotifyAuth a SpotifySearch, ktoré sú zodpovedné za autentifikáciu a vyhľadávanie piesní pomocou Spotify API.

Externé knižnice: Aplikácia využíva knižnice okhttp a gson na spracovanie HTTP požiadaviek a JSON dát.

## 

## 3.2. Hlavný prúd vykonávania

Vstupným bodom aplikácie je trieda Main, ktorá obsahuje metódu main. Táto metóda slúži na zobrazenie menu v konzole a spracovanie vstupov od užívateľa. Tu je opis základnej logiky vo vnútri metódy main:

Inicializácia komponentov: Vytvoria sa inštancie InMemorySongDAO a MusicLibraryService.

Užívateľské menu: Aplikácia vypíše možnosti menu a načíta výber užívateľa.

Spracovanie príkazov: Na základe vstupu užívateľa sa vykoná daná funkcia, od vyhľadania pesničky pomocou Spotify API po manažovanie playlistov.

Neustále opakovanie (loop): Menu sa zobrazuje opakovane po každej operácii, čo užívateľovi umožňuje vykonávať ďalšie akcie, kým sa rozhodne ukončiť program.

## 3.3. Funkcie správy dát

Pridanie piesne: Užívateľ zadá názov, interpreta a žáner (voliteľný). Aplikácia overí, či pieseň už bola pridaná, a potom vyhľadá dostupné piesne cez Spotify API. Užívateľ si vyberie želanú pieseň, ktorá sa pridá do lokálnej databázy.

Odstránenie piesne: Užívateľ zadá názov piesne na odstránenie. Aplikácia overí jej existenciu a odstráni ju.

Správa playlistov: Užívatelia môžu vytvárať playlisty, pridávať do nich piesne, odstraňovať piesne z playlistov a prezerať si playlisty.

Vyhľadávanie: Užívatelia môžu vyhľadávať piesne podľa názvu (artist, song name) a playlisty podľa mena.

## 3.4. Výstup a užívateľské rozhranie

Všetka interakcia s užívateľom prebieha cez konzolové rozhranie, kde sú výstupy zobrazené v texte a užívateľ zadáva svoje príkazy cez klávesnicu. Toto jednoduché rozhranie umožňuje rýchlu a efektívnu manipuláciu s dátami bez potreby grafického užívateľského rozhrania.

## 3.5. Integrovanie s externým API

Aplikácia sa spolieha na Spotify API pre získavanie informácií o piesňach:

Autentifikácia: SpotifyAuth získava prístupový token používaním clientID a clientSecret, ktoré sú uložené v konfiguračnom súbore (config.properties).

Vyhľadávanie piesní: SpotifySearch realizuje požiadavky na vyhľadávanie piesní na základe užívateľom zadaného názvu a interpreta. Výsledky z Spotify sú spracované a prezentované užívateľovi na výber.

## 3.6. Spracovanie chýb

Aplikácia obsahuje základné spracovanie chýb, aby sa zabezpečilo, že užívateľ bude informovaný o problémoch, ako sú chyby v sieťovom pripojení alebo chyby pri spracovaní dát:

Výnimky pri sieťových požiadavkách: Ak dôjde k chybe pri komunikácii so Spotify API, užívateľ je informovaný o probléme a operácia sa ukončí.

Chyby pri vstupoch od užívateľa: Neplatné alebo neexistujúce vstupy sú zachytené a užívateľ je vyzvaný, aby zadal platné údaje.

## 3.7. Bezpečnosť a správa citlivých údajov

Citlivé údaje, vrátane prihlasovacích údajov k Spotify, sú bezpečne uložené v konfiguračnom súbore, ktorý je načítaný za behu a nie je priamo prístupný z kódu, čo minimalizuje riziko ich úniku.

## 3.8. Rozšíriteľnosť a údržba

Kód aplikácie je štruktúrovaný tak, aby bol ľahko rozšíriteľný a udržiavateľný. Logika spracovania dát je oddelená od užívateľského rozhrania, čo uľahčuje zmeny bez nutnosti zasahovať do kódu rozhrania. Taktiež je pripravený na pridanie ďalších funkcionalít.

# 4 Záver

Vytvorenie tejto aplikácie predstavovalo významný krok v mojom porozumení práci s externými API a vývoji softvérových aplikácií vo všeobecnosti. Počas vývoja som získal cenné skúsenosti s programovacím jazykom Java, prácou s knižnicami pre HTTP požiadavky a spracovanie JSON dát. Aplikácia úspešne demonštruje, ako je možné využiť verejné API, ako je Spotify, na vytvorenie užitočného nástroja pre milovníkov hudby, ktorý umožňuje vyhľadávanie a správu hudobných zbierok. Tento projekt taktiež poukázal na dôležitosť bezpečného spracovania a uchovávania citlivých údajov, ako sú prihlasovacie údaje k API. Ďalším krokom v rozvoji projektu by mohlo byť pridanie grafického užívateľského rozhrania, ktoré by zlepšilo interakciu s aplikáciou a urobilo ju prístupnejšou širšiemu okruhu užívateľov. Napokon, tento projekt nielen zvýšil moje technické zručnosti, ale tiež posilnil moju schopnosť analyzovať a implementovať komplexné softvérové riešenia, čo je zručnosť, ktorú si nesmierne cením na mojej ceste stať sa úspešným programátorom.