

VYSOKÉ UČENÍ TECHNICKÉ V BRNE

FAKULTA INFORMAČNÝCH TECHNOLOGIÍ

Internetové aplikace

Webová aplikace pro zpracování a prezentaci statistik z
aplikace Spotify

23. dubna 2023

Tým xnosko05

Eva Moresová xmores01
Daša Nosková xnosko05

Obsah

1	Úvod	2
2	Návrh	2
2.1	Server	2
2.1.1	Autorizácia	2
2.2	App	3
2.2.1	Top Songs	3
2.2.2	Artist Meter	3
2.2.3	Music DNA	3
2.2.4	Playlist Timeline	4
2.2.5	Discover	4
2.2.6	Odhlásenie	4
3	Implementácia	4
3.1	Použité knižnice	4
3.2	Autorizácia	5
3.2.1	Spravovanie tokenov	5
3.3	Odhlásenie	5
3.4	Top Songs	5
3.5	Artist Meter	5
3.6	Music DNA	5
3.7	Playlist Timeline	6
3.8	Discover	6
4	Spustenie aplikácie	6
5	Štruktúra zložiek	7
6	Konfiguračný súbor	7
7	Rozpis práce	7

1 Úvod

Cieľom projektu bolo vytvoriť webovú aplikáciu pre spracovanie a grafickú prezentáciu štatistík a analytík založených na dátach užívateľských účtov z aplikácie Spotify. Pre získanie dát bolo využité verejné API poskytované službou Spotify¹. Pre projekt boli zvolené technológie ReactJS a JavaScript. Pre používanie aplikácie je nutné mať vytvorený Spotify účet.

2 Návrh

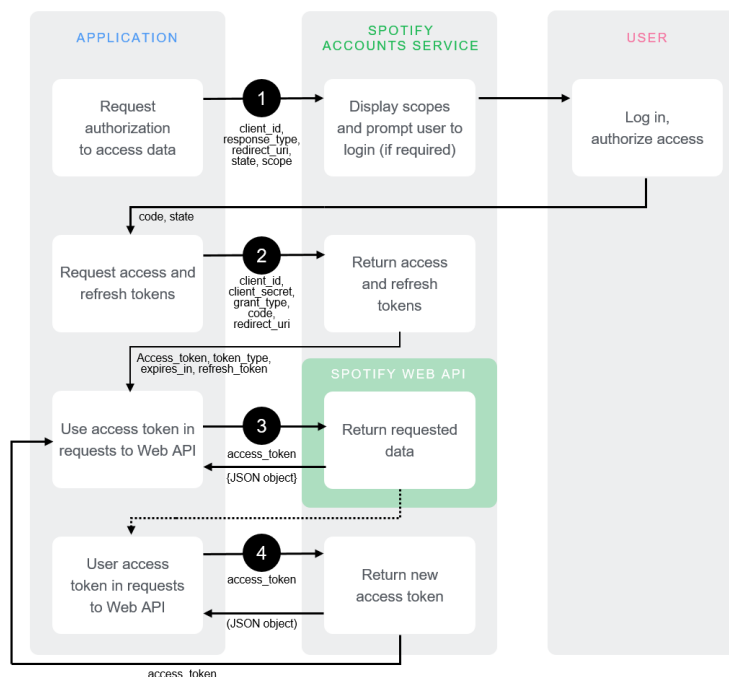
Kompletná aplikácia je zložená z 2 častí - app a server. Server slúži pre autentizáciu užívateľov a získavanie tokenu používaného pre získavanie dát. V časti app sa nachádza implementovaná aplikácia, ktorá získava, spracováva a zobrazuje dáta.

2.1 Server

Pre implementáciu serveru je využitý framework Express a taktiež knižnica Cors a Axios pri žiadostach o dáta. Server počúva na porte 3001. Server poskytuje 4 nasledujúce endpointy:

- GET /auth - autentizácia aplikácie.
- GET /callback - callback na ktorý sa presmeruje po potvrdení autentizácie. Vracia prístupový a refresh token.
- GET /token - vracia prístupový a refresh token.
- GET /refresh - obnovenie prístupového tokenu na základe refresh tokenu.

2.1.1 Autorizácia



Obrázek 1: Diagram authorization code flow (prevzaté z developer.spotify.com)

¹<https://developer.spotify.com/documentation/web-api>

Prvým krokom k používaniu aplikácie je prihlásenie do užívateľského účtu aplikácie Spotify. Autorizácia prebieha cez OAuth a bol využitý prístup Authorization Code Flow (viď 1), pri ktorom užívateľ autorizuje aplikáciu iba raz. Po potvrdení autorizácie, je užívateľ presmerovaný na privítaciu stránku a môže si zobrazit' štatistiky.

2.2 App

Aplikácia je implementovaná pomocou frameworku ReactJS a počúva na porte 5173. Aplikácia je zložená zo 7 stránok:

- `/login` - Prihlasovacia stránka.
- `/` - Privítacia stránka zobrazená po prihlásení.
- `/topsongs` - Zobrazenie najpočúvanejších piesní.
- `/artists` - Top najpočúvanejších interpretov.
- `/music-dna` - Analýza vlastností najviac počúvaných piesní.
- `/playlist-timeline` - Zobrazenie časovej osi vydania piesní v playliste.
- `/discover` - Odporúčanie nových piesní podľa dekády.

2.2.1 Top Songs

Stránka zobrazuje 5 najpočúvanejších piesní v 3 časových obdobiach - za celý čas používania aplikácie, za posledných 6 mesiacov, za posledné 4 týždne.

2.2.2 Artist Meter

Roztriedenie top 50 najpočúvanejších interpretov do kategórií na základe ich celkovej popularity a priradenie užívateľovi, na základe priemernej popularity interpretov, charakteristiku jeho štýlu. Kategórie sú rozdelené do nasledujúcich intervalov podľa popularity:

- $0.0 \geq \text{popularita} < 30.0$
- $30.0 \geq \text{popularita} < 50.0$
- $50.0 \geq \text{popularita} < 75.0$
- $75.0 < \text{popularita}$

2.2.3 Music DNA

Analýzou vlastností 100 najpočúvanejších piesní za celé obdobie používania aplikácie je užívateľ zaradený do jednej z šiestich kategórií:

- Dreamy wanderer
- Electric explorer
- Rhythmic rebel
- Soulful sage
- Brooding storm
- Musical maverick

Na stránke je zobrazený popis kategórie, charakteristika človeka, ktorý do tejto kategórie spadá, graf priemerných hodnôt pozorovaných parametrov piesní a 3 z najpočítavanejších piesní, ktoré túto kategóriu reprezentujú. Design stránky a použitá kombinácia farieb odráža povahu danej kategórie. Pri analýze piesní sú brané do úvahy parametre:

- danceability - ako sa hodí pesnička na tancovanie
- energy - energickosť piesne (intenzita a aktivita)
- valence - pozitivita piesne
- tempo - tempo v BPM
- acousticness - akustickosť piesne

2.2.4 Playlist Timeline

Užívateľ si z ponuky vyberie jeden zo svojich playlistov (na začiatku je zvolený prvý zo zoznamu, prístupných je maximálne 5). Na stránke sa potom zobrazí časová os zachytávajúca dátumy vydania jednotlivých piesní, ktoré playlist obsahuje. Užívateľ sa po tejto osi môže posúvať oboma smermi pomocou tlačítek.

2.2.5 Discover

Odporúčanie nových piesní, z rôznych dekád od roku 1950, na základe užívateľovej hudobnej preferencie podľa analýzy top 50 piesní za posledných 6 mesiacov a ich nasledujúcich parametrov danceability, energy, valence, tempo popísané v 2.2.3 a instrumentalness, čo je zastúpenie vokálov (1 - bez vokálov).

Stránka začína tlačítkami, z ktorých si užívateľ môže vybrať dekádu pre ktorú chce objaviť nové piesne. Po kliknutí na tlačítko sa zobrazí 5 odporúčaných piesní z danej dekády. Užívateľ si môže zobrazit ďalšie piesne z danej dekády, prejsť na nasledujúcu dekádu alebo sa vrátiť na počiatočnú stránku so všetkými tlačítkami. Po kliknutí na srdiečko pri piesni je užívateľ presmerovaný na stránku Spotify pre daný album. V prípade prejdienia kurzorom po riadku príslušiacemu danej pesničke, začne pesnička hrať, po odídení s kurzorom pieseň hrať prestane.

2.2.6 Odhlásenie

Užívateľ sa môže odhlásiť kliknutím na príslušný piktogram zobrazený v pravom hornom rohu stránky.

3 Implementácia

3.1 Použité knižnice

Pri implementácii boli použité nasledujúce knižnice a balíčky:

- axios - knižnica na vykonávanie požiadaviek na API.
- body-parser - balíček pre parsovanie tela requestov.
- cookie-parser - balíček pre parsovanie HTTP request cookies.
- cors - balíček poskytujúca middleware pre umožnenie cross-origin zdieľanie.
- dotenv - balíček pre načítanie enviromentálnych premenných zo súboru `.env`.
- express - knižnica pre tvorbu serveru a API.
- react - knižnica pre tvorbu UI
- react-apexcharts - knižnica pre vykresľovanie grafov v knižnici React.
- react-dom - React balíček pre prácu s DOM.
- react router dom - balíček pre presmerovanie v knižnici React.

3.2 Autorizácia

Po stlačení tlačítka LOGIN je poslaná žiadosť na server `/auth`, ktorý pošle žiadosť na `authorize` endpoint stránky Spotify spolu s `clientID`, `clientSecret`, príslušnými povoleniami (`scopes`) a `callback uri`. Po povolení od užívateľa sa žiadosť presmeruje na priloženú `callback URI` `/callback`, kde sa overí stav, že všetko prebehlo v poriadku a pošle sa žiadosť na získanie tokenu od Spotify. Server nastaví prístupový a refresh token a presmeruje užívateľa na stránku aplikácie spolu s prístupovým tokenom, ktorý sa v aplikácii uloží do lokálneho úložiska, kde je ďalej využívaný komponentom `Auth`, ktorý obaluje jednotlivé stránky a povoľuje k nim prístup.

3.2.1 Spravovanie tokenov

Po úspešnom prihlásení je prístupový token uložený okrem lokálneho úložiska aj do hlavičiek inštancie `axios`, pomocou ktorej sú následne odosielané všetky požiadavky. Obnovenie prístupového tokenu je realizované pomocou `axios` interceptorov. V prípade, že na žiadosť odoslanú autorizovanou inštanciou `axios` príde odpoveď so statusom `Unauthorized`, je poslaná žiadosť na server `/refresh` o nový prístupový token, aktualizuje sa hodnota tokenu v lokálnom úložisku a hlavičke inštancie `axios` a zopakuje sa pôvodná žiadosť. V prípade, že vypršala aj platnosť refresh tokenu na serveri, nebude žiadosť o nový prístupový token úspešná a dojde k odhláseniu užívateľa.

3.3 Odhlásenie

Odhlásenie je zastúpené komponentom `Logout`, ktorý vymaže token z lokálneho úložiska a užívateľa presmeruje na prihlasovaciu stránku.

3.4 Top Songs

Stránka `/topsongs` (`TopSongsPage`) je zložená z detského komponentu `Songs`, ktorý pre každý časový úsek využíva ďalší komponent `TopSongsCard`. Tento komponent získa pomocou `get` žiadosti top pesničky v zadanom časovom období, cez funkciu `getTopTracks` a následne tieto dáta zobrazí.

3.5 Artist Meter

Stránka `/artists` (`ArtistMeterPage`) využíva detský komponent `ArtistMeter`, ktorý pomocou funkcie `getArtists` získa top najpočítanejších interpretov za celý čas používania aplikácie Spotify a vypočíta priemernú popularitu všetkých interpretov. Na základe priemernej popularity je vypísaná charakteristika užívateľa s využitím funkcie `getArtistMeterText`. Okrem charakteristiky stránka zobrazuje rozdelenie týchto top interpretov do kategórií, k čomu využíva komponent `ArtistMeterCategories`, ktorému predáva pole s popularitou a už získané dáta. Tento komponent roztriedi cez funkciu `categorizeByPopularity` pole s popularitou do kategórií podľa intervalov spomenutých vyššie (viď 2.2.2). Na základe objektu s vytvorenými kategóriami sa vytvorí funkciu `createObjForRender` objekt, ktorý každej kategórii priradí priemernú popularitu, fotku najpočítanejšieho interpreta z danej kategórie a počet interpretov v danej kategórii. Tento objekt je následne použitý pri zobrazovaní jednotlivých kategórií.

3.6 Music DNA

Stránka `/music-dna` (`MusicDnaPage`) využíva funkciu `Stránka.getDnaCategory` na získanie kategórie, do ktorej užívateľ patrí. V tejto funkcii sa najskôr požiadajú o zoznam 100 najpočítanejších piesní užívateľa a pre každú sa načíta zoznam parametrov. Pre vybrané parametre, ktoré sú v analýze použité je vypočítaná priemerná hodnota. Priemery sú, ak je potrebné, normalizované (napr. v prípade tempa) a následne je vytvorená charakteristika užívateľa, kde každý sledovaný parameter nadobúda jednu z hodnôt:

- low - priemerná hodnota < 0.3
- moderate - priemerná hodnota ≥ 0.3 a ≤ 0.6
- high - priemerná hodnota > 0.6

Každá kategória má pre parametre definované pole hodnôt, ktoré sú pre ňu prijateľné. Kategórie sú filtrované podľa vytvorenej charakteristiky užívateľa a výsledok určuje správnu kategóriu. Môže sa stať, že pre danej charakteristike neodpovedá žiadna z definovaných kategórií, keďže kombinácia parametrov definovaná pre kategórie nie je vyčerpávajúca. V tomto prípade je použitá kategória Musical Maverick. V prípade, že charakteristika odpovedá viacerým kategóriám, je vybraná vždy prvá v poradí.

Pre každú kategóriu je okrem popisu zobrazený aj graf parametrov užívateľovej charakteristiky. Veľkosť stĺpca odpovedá (normalizovanej) priemernej hodnote parametra z charakteristiky. Zo zoznamu 100 najpočítavanejších piesní sú potom vybrané prvé 3, ktorých parametre odpovedajú parametrom definovaným pre zvolenú kategóriu.

3.7 Playlist Timeline

Stránka `/playlist-timeline` (`PlaylistTimelinePage`) požiada o zoznam playlistov užívateľa funkciou `getUserPlaylists`. Z nich je buď vybraný prvý alebo môže užívateľ zvoliť inú hodnotu pomocou dropdownu, ktorý je implementovaný komponentom `Dropdown`. Výberom playlistu sa spustí zavolanie funkcie `getTimelineItems`, ktorá požiada o piesne playlistu, extrahuje z odpovede dátumy vydania piesní a usporiada podľa nich výsledný zoznam. Piesne sú potom zobrazené pomocou `Timeline` na časovej osi.

3.8 Discover

Stránka `/discover` (`DiscoverPage`) požiada o analýzu top počúvaných piesní cez funkciu `getFeatures()`, ktorá vráti objekt získaných vlastností spomenutých v 2.2.5. Vo funkcii `getFeatures()` sa získajú vlastnosti všetkých piesní a následne sa vypočítajú ich minimálne, maximálne alebo cieľové hodnoty na základe 25% a 75% kvantilov, popr. ich priemeru a priradia sa ku klúčom, ktoré odpovedajú požiadavkam na parametre nasledujúceho volania Spotify API. Tento objekt je predaný komponentu `TimeCapsule`, ktorý objekt predá ďalšiemu komponentu `Recommendations` spolu so získanými seeds pre piesne vo vybratej dekáde. Seeds sa získajú volaním funkcie `search`, ktorá cez Spotify API vyhľadá piesne z danej dekády. Komponent `Recommendations` vyberie náhodných 5 seeds (limit spotify API) a spolu s pripraveným objektom s vlastnosťami ich predá funkcii `recommendations`, ktorá vráti odporúčané piesne. Keďže sa nedá pri hľadaní odporúčaných piesní nastaviť filter podľa rokov vydania, výsledky funkcie `recommendations` sa vyfiltrujú podľa žiadaných rokov (`filterYear`). V prípade ak sa vo výsledkoch (uložené v poli `recommended`) nenachádza dostatočný počet piesní, konkrétne 5, tak sa tento proces opakuje. Užívateľ môže požiadať aj o ďalšie piesne, a v takomto prípade sa využíva nastavenie premennej `refresh` cez `setRefresh` a `useEffect`, ktorý predchádzajúci výsledok oreže o prvých 5 piesní. Ak predchádzajúci výsledok (`recommended`) obsahuje stále 5+ piesní, tak sa použijú už načítané piesne. V opačnom prípade sa opäť opakuje proces s výberom náhodných seedov, získaním odporúčaní a filtráciou podľa roku. Prehrávanie piesní je implementované cez komponent `AudioPlayer`, v ktorom je obalený riadok príslušiaci danej piesni.

4 Spustenie aplikácie

Pre spustenie aplikácie je nutné aplikáciu registrovať na stránke <https://developer.spotify.com/dashboard> a získať `client id` a `client secret`, ktoré sa uložia do konfiguračného súboru `.env` (6). Po registrácii aplikácie je nutné na Spotify Dashboard registrovať užívateľov a nasledujúce endpointy:

- `http://localhost:5173/`
- `http://localhost:5173/callback`
- `http://localhost:3001/callback`

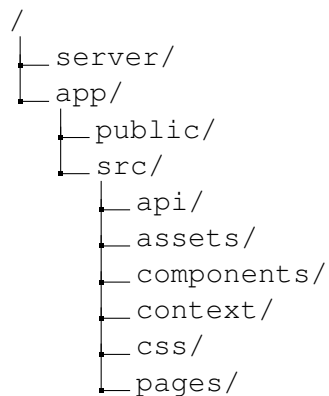
Lokálne spustenie server:

```
cd server
npm install
npm start
```

Lokálne spustenie app:

```
cd app
npm install
npm run dev
```

5 Štruktúra zložiek



Obrázek 2: Štruktúra zložiek

6 Konfiguračný súbor

Listing 1: Ukážka štruktúry konfiguračného súboru .env

```
1 CLIENT_ID= XXXX
2 CLIENT_SECRET=XXXX
3 REDIRECT_URI=http://localhost:3001/callback
4 APP_URL=http://localhost:5173/
```

7 Rozpis práce

Daša Nosková:

- aplikácia server, autorizácia
- stránka Login, Welcome, Top Songs, Artist Meter, Discover
- dokumentácia

Eva Moresová:

- spravovanie tokenov
- stránka Music Dna, Playlist Timeline
- dokumentácia