

Osztályok

Store:

- albums: *Array<Album>*
- Constructor(): *Store*
- addAlbum(Obj)
- formatJSON(JsonObj inputAlbum): *Album*
- checkInput(JsonObj inputAlbum): *?Error*
- getAll(): *Array<Album>*
- getBestOf(Int id, Int number): *String*
- getAlbum(Int id): *Album*

Album:

- id: *ID*
- tracks: *ARRAY<Track>*
- bestSongFinder: *BestSongFinder*
- Constructor(Store store, Obj albumData): *Album*
- getBests(Int number): *Array<Track>*

Track:

- id: *ID*
- title: *String*
- frequency: *Int*
- zipfIndex: *Int*
- Constructor(Album album, String title, Int frequency): *Track*
- countZipfIndex()

BestSongsFinder:

- album: *Album*
- Constructor(Album album): *BestSongFinder*
- sortByZipf(Album album): *Album*
- findBests(Int number): *Array<Track>*

Az elvárások

- Lehesse megadni az alkalmazásnak albumokat, a rajtuk található számok címével és azok hallgatási gyakoriságával.
- Le kell tudni kérdezni az alkalmazástól egy album n darab legjobb minőségi indexével rendelkező számának címét.
- Ha két szám azonos minőségi indexszel rendelkezik, akkor soroljuk előrébb az albumon is előbb található számot (feltehetőleg okkal tették a szerzők ebben a sorrendben az albumra őket)
- Az előbb definiált műveletekre biztosítson az alkalmazás egy HTTP alapú API-t, az alábbi endpointokkal:
 - POST /albums - egy új album adatainak feltöltésére
 - GET /albums/:album_id/best?top= n - egy album legjobb n számának lekérdezésére
- Minden HTTP API endpoint JSON adattípust várjon és azzal válaszoljon. A küldött és fogadott JSON objektumok szerkezete megtalálható a példánál.
- Az API-n ellenőrizzük a bemenet helyességét, és válaszoljunk 400-as HTTP status code-dal ha hibát találunk a bemenetben (pl. a legjobb n szám lekérdezésénél nincs megadva n).
- A GET-es endpoint legyen idempotens művelet, azaz egymás utáni többszöri végrehajtása járjon ugyanazzal az eredménnyel. Ne módosítson állapotot az alkalmazásban.
- Készüljenek end-to-end tesztek a HTTP API-hoz.

Feltehetjük, hogy egy albumon az első szám meghallgatásának gyakorisága nagyobb, mint nulla.

A feladathoz néhány lépésben segítséget adunk. Érdemes ezt a sorrendet követni (de nem kötelező), így kis lépésekben haladhatunk a megoldás felé. A lépéseknél nem hangsúlyozzuk ki a teszteket, azonban elvárás, hogy az egyes funkcionalitások le legyenek fedve teszttel - ha lehetséges ezek TDD-vel készüljenek el.

Az alkalmazás kódja legyen könnyen olvasható, magától értetődő, kövesse a clean code elveit. A kommenteket kerüljük.