

Légitársaságok 2.0

Backend feladat

Feladat leírás

Légitársaságok járatokat üzemeltetnek városok között. A feladat a legkisebb létszámú városból indulva megtalálni a legrövidebb utat (km-ben) a legnagyobb városba (Dijkstra algoritmus implementálása).

A városok és járatok adatait a két mellékelt (input_city.txt, input_flight.txt) fájl tartalmazza. A megoldás során egy REST MicroService kialakítása szükséges, mely egy memória adatbázissal (H2) dolgozik, az alábbi felépítés szerint.

Felépítés

Airline - légitársaság

- név

City - város

- név
- populáció

Flight - repülőjárat

- Airline - légitársaság
- City - indulási város
- City - érkezési város
- távolság - a két város közti távolság (km - int)
- menetrend - repülési idő (perc - int)

Megvalósítás

Mindhárom adatbázis tábla esetén az alap CRUD műveletek (save, delete, findAll, findById) legyenek megvalósítva, emellett a feladat megoldásához tetszőleges egyéb művelet is felvehető.

Az alkalmazás a localhost:8080 porton induljon el és mindhárom osztály esetében az alábbi Endpoint-okat tartalmazza a CRUD műveletek kiszolgálásához. Az adatok JSON formátumban legyenek küldve és fogadva (kivéve fájlok).

Az <entityName> helyére az alábbi elnevezések kerüljenek:

airline, city, flight

REST Endpoints

```
/<entityName> - GET findAll  
/<entityName>/<id> - GET findById  
/<entityName> - POST save  
/<entityName>/<id> - DELETE delete
```

Az adatok bevitele fájl beküldéssel valósuljon meg.

Az algoritmus eredménye egy List<Flight> objektum legyen, melyben a kiindulási városból az érkezési városba vezető járatok szerepelnek. Az algoritmus legyen futtatható az összes

repülőjáratokkal és csak egy légitársaság járatainak igénybevételével is. Ha a felhasznált járatokkal nincs út a két város között, akkor az eredmény egy üres lista legyen.

Legyen lekérhető légitársaságonként a hozzájuk tartozó járatok listája.

Legyen lekérhető a repülőjáratok listája, melyek két megadott város között járnak. A listában szerepeljenek az oda és vissza induló járatok is.

Ezekhez CRUD Endpoint-ok az alábbiakkal legyenek kiegészítve.

`/upload/city` - POST városok adatai

`/upload/flight` - POST járatok adatai

`/result` - GET eredmény az összes járattal

`/result/{id}` - GET eredmény az adott légitársaság járataival

`/airline/{id}/flight` - GET az adott légitársasághoz tartozó járatok

`/flight/way` - GET két város közötti járatok

Egyéb

A REST Endpoint-ok tesztelésére és a fájlok beküldésére pl. az ingyenesen elérhető Postman program használható.

A program írása közben feltételezhetjük, hogy a beérkező adatok helyesek, így ezek külön ellenőrzése nem szükséges.

Tesztek írása nem elvárás.