Dokumentáció – Node.js Express szerver

Ez az alkalmazás egy egyszerű utazáskezelő API-t valósít meg, ahol a felhasználók regisztrálhatnak, bejelentkezhetnek, majd utazásokat hozhatnak létre, szerkeszthetnek és törölhetnek.

Segítségnek minden ki van kommentelve a filokban is

Fontos: az adatbázis neve: travel\_agency, a táblák automatikusan jónnek létre

Főbb technológiák

- Node.js, Express.js – webszerver

- Sequelize – MySQL adatbázis-kezeléshez

- JWT – hitelesítéshez

- dotenv – környezeti változók kezeléséhez

1. API végpontok (server.js)

Regisztráció

- POST /register

- Bemenet: registerUsername, registerPassword

- Kimenet: sikeres vagy hibás regisztráció

Bejelentkezés

- POST /login

- Bemenet: loginUsername, loginPassword

- Kimenet: JWT token, ha a bejelentkezés sikeres

Saját profil megtekintése

- GET /users/me

- Hitelesítés szükséges (JWT token)

- Visszaadja a bejelentkezett felhasználó adatait

2. Adatbázis struktúra (dbHandler.js)

Felhasználók (users tábla)

|  |  |  |
| --- | --- | --- |
| Mező | Típus | Leírás |
| id | integer | Elsődleges kulcs, automatikus |
| username | string | Egyedi, nem lehet null |
| password | string | Nem lehet null |

Utazások (trips tábla)

|  |  |  |
| --- | --- | --- |
| Mező | Típus | Leírás |
| id | integer | Elsődleges kulcs, automatikus |
| name | string | Egyedi, nem lehet null |
| destination | string | Nem lehet null |
| accommodation | string | Szállás |
| transport | string | Közlekedés |
| description | text | Leírás |
| creator | date | Dátum |
| date | string | A felhasználó neve |

3. Utazások kezelése

Az alábbi végpontokhoz hitelesítés szükséges, kivéve ahol külön jelezve van.

Új utazás létrehozása

- POST /trips

- Bemenet: utazás adatai (pl. name, destination, stb.)

- Mentés a bejelentkezett felhasználóhoz

Saját utazások lekérdezése

- GET /trips

- Visszaadja az adott felhasználó által létrehozott utazásokat

Egy utazás lekérdezése

- GET /trips/:id

- Csak ha az adott utazást a felhasználó hozta létre

Utazás módosítása (hitelesítés hiányzik, javasolt pótolni)

- PUT /trips/:id

- Bemenet: módosított utazás adatok

- Kimenet: módosítás eredménye

Utazás törlése (hitelesítés hiányzik, javasolt pótolni)

- DELETE /trips/:id

- Kimenet: törlés eredménye

Keresés név alapján

- GET /trips/name/:name

- Visszaadja az adott nevű utazást, ha létezik

4. Hitelesítés

Middleware: Auth()

- Ellenőrzi, hogy a kérés tartalmaz-e érvényes Bearer token-t

- Hibás vagy hiányzó token esetén 401-es válasz

5. Indítás

A szerver a .env fájlban megadott PORT porton indul. Példa .env fájl:

PORT=3000

SECRETKEY=valamiTitkosKulcs

Travel Agency WPF Alkalmazás

Ez az alkalmazás egy WPF-ben készült utazási iroda (Travel Agency) rendszer kliensoldali felülete. A célja, hogy a felhasználók regisztráljanak, bejelentkezzenek, és utazásokat kezeljenek (létrehozás, módosítás, törlés, keresés).

Projekt fő elemei

**MainWindow.xaml.cs**

A belépési ablak, amely kezeli a **regisztrációt** és a **bejelentkezést**.

**Fontos elemek:**

* Registration() (gombnyomásra): új felhasználó regisztrálása.
* LoginAsync() (gombnyomásra): felhasználó bejelentkeztetése. Sikeres bejelentkezés után egy másik ablak (Travel\_agency\_idk\_mainwindow) nyílik meg a JWT tokennel.
* StatusText: felhasználói visszajelzések (siker/hibaüzenet) megjelenítésére.

ServerConnection.cs

Ez az osztály felelős az összes **szerverkommunikációért**

**Fő metódusok:**

* **Felhasználói műveletek:**
  + Registration(username, password)  
    → Felhasználó regisztrációja.
  + LoginAsync(username, password)  
    → Bejelentkezés és JWT token visszakérése.
  + GetProfileDataAsync(token)  
    → Saját felhasználói adatok lekérdezése.
* **Utazáskezelés (Trips):**
  + CreateTrip(...)  
    → Új utazás létrehozása.
  + GetAllTrips(token)  
    → Összes utazás lekérdezése (felhasználónkként).
  + GetTripById(token, id)  
    → Egy adott utazás részletes adatainak lekérése (felhasználónkként).
  + UpdateTrip(token, id, ...)  
    → Egy meglévő utazás szerkesztése.
  + DeleteTrip(token, id)  
    → Egy utazás törlése.
  + SearchTripsByName(token, name)  
    → Utazások keresése név alapján (felhasználónkként).

**Technológia:**

* **HTTPClient**
* **Json.NET (Newtonsoft.Json)**
* **JWT token**: hitelesítéshez

Projekt Részei

* MainWindow.xaml.cs – regisztráció, login kezelése
* Travel\_Agency\_mainwindow.xaml.cs – felhasználói felület, ahol lehet utazásokat menaggelni. (minden ki van kommentelve, ami fontos)
* ServerConnection.cs – Szerverrel való kommunikációért felelős, mind a MainWindow, és a Travel\_Agency\_mainwindow ezt használja

Credits:

* Potyondi Zsombor – 3. feladatrész
* Komjáti Gábor – 2. feladatrész
* Lengyel Dániel – 1. feladatrész

github link a wpf-hez: https://github.com/LD12I/Travel\_agency\_project\_FixedVersion