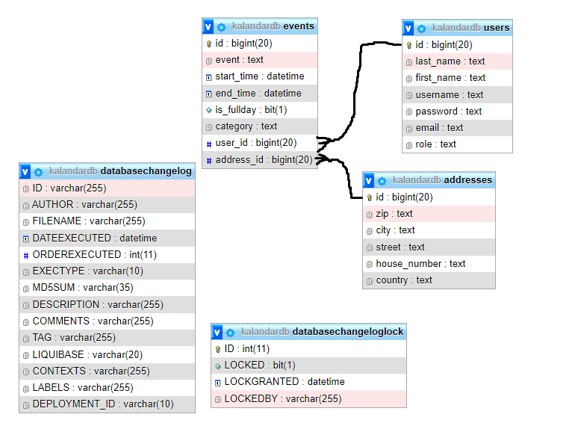
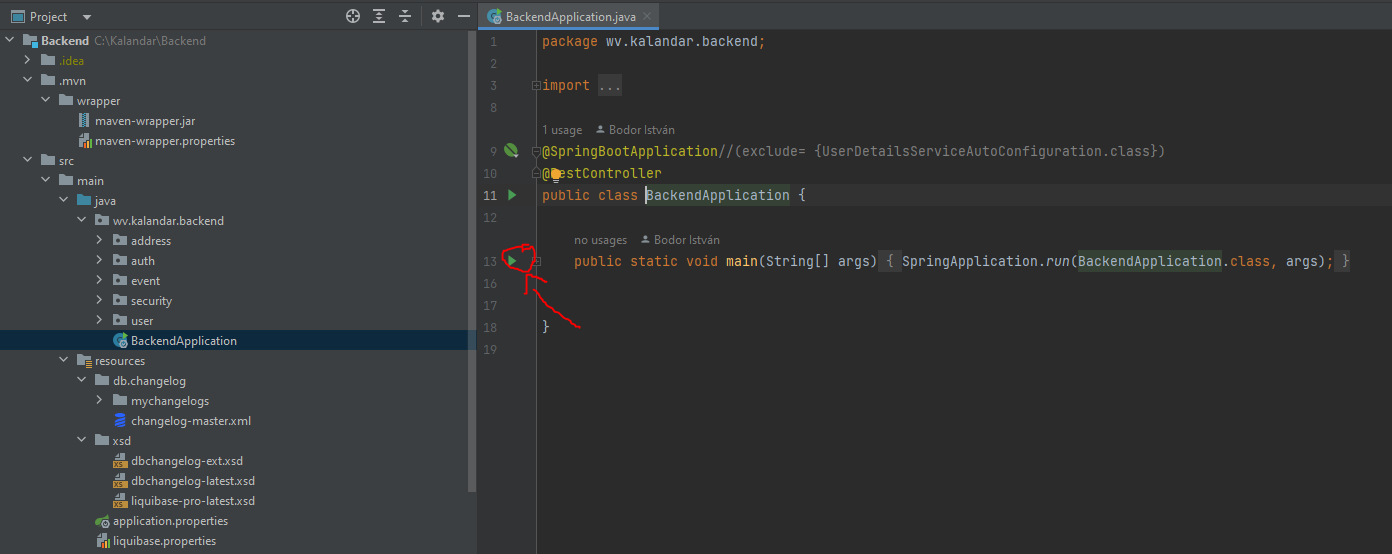
**Kalandár backend-db fejlesztői dokumentáció**

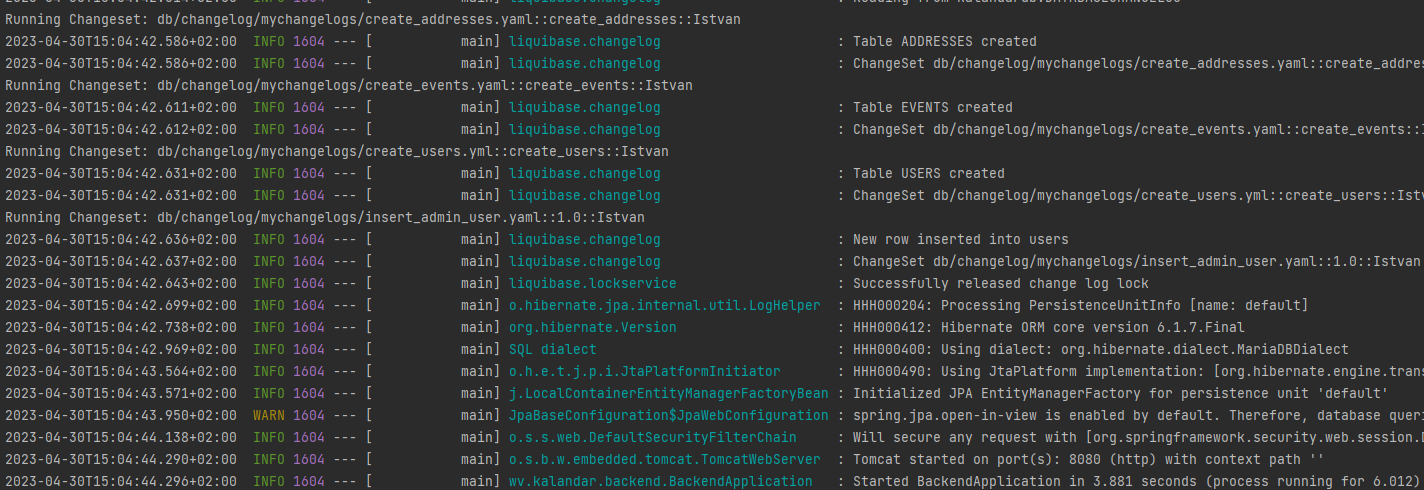
**+Tesztek**

Adatbázis-diagram: 

**ADATBÁZIS – FONTOS!** Mellékelve adatbázis dump file nem található, mivel a táblákat és a dummy adatokat a Liquibase hozza létre a backend alkalmazás indításakor. Egyetlen kritérium az üres „**kalandardb**” nevezetű adatbázis megléte az alkalmazás indítása előtt!

Alkalamzás indítása (IntelliJ IDEA):



Amennyiben létezik a kalandardb nevezetű üres adatbázis létrejönnek a tábláink és elindul az alkalmazás a 8080-as porton.

**Alkalmazás, végpontok leírása**

Alapméretezett admin felhasználó:

email: admin@admin.ka

pw: admin

Neki van egyedül jogosultsága a /api/admin kezdetű végponthoz, legyen szó bármilyen kérésről.

A továbbiakban ő fog tudni admin ROLE-t osztani a többi felhasználónak.

Autentikáció:

REGISZTRÁCIÓ

/api/auth/register

Egy felhasználót lehet regisztrálni amennyiben nem létezik még az email címe. Authenctication header NEM szükséges.

Nem lehet admin role-t adni alapméretezetten, szóval bármit adunk meg USER-role lesz mentve a felhasználónak.

CSAK POST kérés.

Példa:

POST http://localhost:8080/api/auth/register

Content-Type: application/json

{

"firstName": "test",

"lastName": "user",

"username": "testusername",

"role": "USER",

"email": "testemail",

"password": "testpassword"

}

BEJELENTKEZÉS/REFRESH TOKEN

/api/auth/login

Helyes, létező email és jelszó páros megadása esetén vissza kapunk egy JWT tokent.

CSAK POST kérés.

Példa:

POST http://localhost:8080/api/auth/login

Content-Type: application/json

{

"email": "testemail",

"password": "testpassword"

}

Válasz példa:

{

"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXN0ZW1haWwiLCJpYXQiOjE2Nzk0NzI0NzUsImV4cCI6MTY3OTQ3MzkxNX0.BRtohNE2l7ar7IRGdbVXWXjoT7XGmDsVd9MHLQFEw\_w"

}

A token 1 éven keresztül él, utána refresh token kérése szükséges amit megkapunk egy új login kérés küldésével.

Az összes többi végpont eléréséhez AUTORIZÁCIÓ SZÜKSÉGES.

A kérésnek tartalmaznia kell egy Authorization HEADERT.

FONTOS, hogy az authorization header a „Bearer „ kifejezéssel kezdődjön, utána egy szóköz után következzen a JWT token.

Példa:

POST <http://xyz>:0000/endpoint

Content-Type: application/json

Authorization: Bearer {JWT TOKEN}

EVENT végpontok

Összes event lekérése.

Példa:

GET <http://localhost>:8080/api/events

Authorization: Bearer {JWT}

Egy userhez tartozó eventek lekérése.

Példa:

GET <http://localhost>:8080/api/events/user/{existingUserId}

Authorization: Bearer {JWT}

Új event felvétele:

Ha a cím nem létezik az ADDRESSES táblában bekerül oda, és az esemény megkapja a megfelelő cím azonosítót.

Ha a cím létezik az ADDRESSES táblában megkapjuk a létező cím id-t és mégegyszer nem kerül mentésre a cím.

A dátum formátuma fontos illetve, hogy létező user id-t adjunk meg.

Amennyiben az event megegyezik egy, már létező event leírásával illetve a kezdés időpontja is azonos az event nem vehető fel.

Az event kezdés időpontja kötelező!

POST <http://localhost>:8080/api/events

Content-Type: application/json

Authorization: Bearer {JWT}

{

„event”: „Az event leírása.”,

„startTime”: „2023-03-10T12:00”,

„endTime”: „2023-03-10T12:30”,

„fullDay”: false,

„category”: „testcategory”,

„address”: {

„city”: „Budapest”,

„zip”: „1234”,

„country”: „Magyarország”,

„street”: „Teszt utca”,

„houseNumber”: „123/abc”

}

}

Létező event törlése:

DELETE <http://localhost>:8080/api/events/{existingEventId}

Authorization: Bearer {JWT}

Létező event módosítása:

Módosíthatjuk az összes elemét egyszerre az eseménynek, ugyan akkor egyszerre csak/néhány egy elemet is.

Üres értékre módosítás nem lehetséges.

PUT <http://localhost>:8080/api/events/{existingEventId}

Content-Type: application/json

Authorization: Bearer {JWT}

{

„event”: „Updated event”,

„startTime”: „2023-10-10T12:00”,

„endTime”: „2023-10-10T14:00”,

„category”: „updatedcategory”,

„fullDay”: false,

„address”: {

„city”: „Updated city”,

„country”: „Updated country”,

„zip”: „4321”,

„street:”: „Updated street”,

„houseNumber”: „321/bca”

},

„user”: {

„id”: 2

}

}

USER végpontok

User profil lekérése:

GET <http://localhost>:8080/api/user/profile

Authorization: Bearer {JWT}

User jelszó változtatása:

Egy password String paraméter passzolásával megváltoztatja a felhasználó jelszavát.

PUT <http://localhost>:8080/api/user/password

Content-Type: application/json

Authorization: Bearer {JWT}

{

„password”: „password”

}

ADMIN végpontok

- Csak akkor lehetséges elérni, ha a felhasználó rendelkezik admin jogosultsággal.

- Admin jogosultságot kezdetben csak az alapméretezett ADMIN felhasználó adhat, regisztrálni adminként nem lehetséges.

Összes felhasználó lekérése:

GET <http://localhost>:8080/api/admin/users

Authorization: {JWT}

Felhasználó törlése:

DELETE <http://localhost>:8080/api/admin/user/{existingUserId}

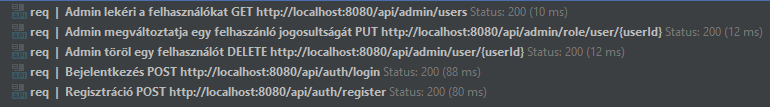
Authorization: Bearer {JWT}

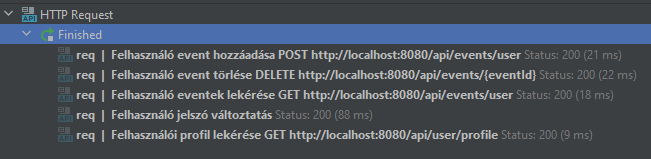
Felhasználó jogosultság változtatása

Ha USER jogosultsága volt admin lesz, ha ADMIN jogosultsága volt USER lesz.

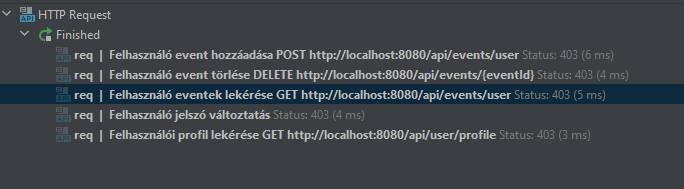
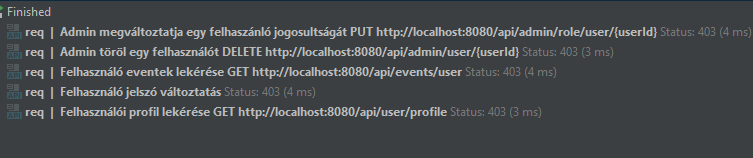
PUT <http://localhost>:8080/api/admin/role/user/{existingUserId}

Authorization: Bearer {JWT}

**Tesztek helyes kérésekkel**



**Tesztek rossz autentikációval**



**Tesztek rossz kérésekkel**