

Rövid technikai leírás az InnoTeq junior tesztfeladatához

A forráskód működését kommenteltem, így azokra a részekre most részletesebben nem térnék ki.

Az alkalmazás Hibernate-tel készült, tehát az inicializáló SQL szkript használata mellőzhető. Az adatbázis táblák és kapcsolataik, jogosultságaik automatikusan létrejönnek a program elindításakor a model osztályok alapján. Ám az SQL szkriptemet az adatfeltöltés tekintetében érdemes megnézni!

Az adatmodellemből kifolyólag sok helyen alkalmaztam CRUD és JPA Repository interfészeken keresztül való kommunikációt az adatbázissal. Azonban figyeltem arra, hogy a biztonság kedvéért adatmódosítást (beszúrás, frissítés, törlés) JPQL-en keresztül végezzek, csak szimpla lekérdezésekhez (SELECT) használtam natív SQL-t.

A beküldött archívumban található fájlok rövid ismertetése:

- A kész Spring projekt mappája, itt az „src/main/webapp/...” alá fel van csatolva a Vue.js frontend is. De mivel a spring projektben a Vue.js frontend „kibuildelt” verziója látszik csak, ezért a Vue.js projekt forráskódját a Spring projekt mellett külön mappában is elküldöm.
- Két ER diagram az adatbázisról. Az egyik, az „eredeti” az a tervezés fázisában készült (draw.io), a másik pedig már a ténylegesen elkészült adatbázist mutatja be (DBeaver-ből exportálva).
- Egy SQL szkript, amely létrehozza a DB-t, és a vége felé vannak benne példák, hogy hogyan lehet új értékeket felvenni ebbe az adatbázisba (Megj.: a „product_info” tábla ahová a termékneveket és egységárait manuálisan kell feltölteni).

Használt technológiák és eszközök:

Backend [IDE: IntelliJ IDEA Ultimate Edition]:
Java, Spring Boot, Hibernate, Lombok, AOP

Frontend [IDE: Sublime Text]:
Vue.js (BootstrapVue), HTML, CSS

Adatbázis [Tool: DBeaver]:
PostgreSQL