

RepMaster

Rétegek megvalósítása, kapcsolata

JPA entitás osztályokban tároljuk az adatokat, melynek példányait relációs adatbázisban perzisztensen tárolja a JPA.

Az entításokra írtunk specifikus repository-kat, ezek az interfészek segítségével az adatelérési kód nagy részét megspóroltuk. Ezeken keresztül történik az adatelérés CRUD műveletekkel.

Az entításokra specifikus service osztályokat is írtunk, itt vannak megvalósítva az üzleti logikát használó függvények, amiket a webréteg használ. A service metódusokra kitettük a `@Transactional` annotációt, így egy tranzakcióba fogjuk a repository metódusokkal végzett műveleteket.

A főbb funkciók (bejelentkezés/regisztrálás, keresés, edzőtermek/edzőtervek/gyakorlatok/eredmények kilistázása, új hozzáadása) külön oldalakon érhetőek el, amik kinézetéért és működéséért thymeleaf-es html-ek, illetve controller osztályok felelnek, amelyek a kérések feldolgozását végzik.

Van egy `@SpringBootApplication` annotációval ellátott `TrackerApplication` singleton főosztály, amivel futtatható maga az adatbázis program.

Adatmodell

- **RegisteredUser**

A RegisteredUser egy regisztrált felhasználót testesít meg, akik megadott felhasználónévvel és jelszóval regisztráltak. Számukra elérhető minden funkció: keresés, saját edzőtermek, edzőtervek, illetve gyakorlatok elmentése, valamint saját eredmények feljegyzése az egyes gyakorlatokból. Ők emellett tudnak is értékelni publikus edzőtermeket, edzőterveket, vagy gyakorlatokat. Az osztályban el vannak tárolva a felhasználóhoz tartozó edzőtermek, edzőtervek, gyakorlateredmények és értékelések.

- **Rating**

A rating osztály testesít meg egy értékelést, amely eltárol 1-5-ig egy értékelés értéket, illetve opcionális string kommentet. Ilyen értékeléseket az olyan osztályokra lehet tenni, amelyekre lehet keresni, illetve csak regisztrált felhasználók hozhatnak létre, illetve törölhetik a sajátjukat. Értékelésekből átlagot is tud számítani.

- **Rateable**

A rateable egy absztrakt osztály, amely jelképezi az értékelhető osztályokat(gym, workout, exercise), hogy ezeket keresésnél egységesen lehessen kezelni. Ezekre az osztályokra be lehet állítani, hogy publikusan elérhetőek legyenek-e, tehát megjelenjen a keresési találatok között, illetve más is használhassa.

- **Gym**

A gym osztályban megtalálhatók egy edzőterem információi, a hozzá tartozó edzőtervek, illetve a split.

- **Split**

Milyen típusú heti beosztást csinál az adott felhasználó egy adott edzőteremben. Eltárolja a split nevét, illetve, hogy ezt heti hány nap csinálja.

- **SplitType**

Belső enum osztály, amelyben fel vannak sorolva, milyen típusú splitek érhetőek el.

- **Equipment**

Az Equipment egy enum osztály, amely egy edzőterem felszereltségi szintjét mutatja.

- **Workout**

A workout osztályban vannak egy edzőterv információi, izomcsoportok, amiket dolgoz, és a gyakorlatok.

- Exercise

Az Exercise osztályban vannak egy gyakorlat információi, elsődleges, illetve másodlagos izomcsoportok, amiket edz, illetve hogy milyen eredmény bejegyzések tartoznak hozzá.

- MuscleGroup

A muscleGroup egy enum osztály, amely tartalmazza az edzhető izmokat.

- ExerciseResult

Az ExerciseResult osztályban vannak egy gyakorlat eredményeinek információi, a dátum, amikor feljegyezték, illetve a szettek.

- Set

A Set osztályban vannak a szett információi, mint az ismétlések száma, a súly, illetve az edzés volumen.

Részletesebb osztályleírások a javadoc-ban található

Tipikus megoldások

- Keresésnél edzőterem/edzőterv/gyakorlat hozzáadása egy felhasználóhoz
A probléma, hogy a keresési találatok egy Rateable listában vannak elmentve, így ha például rákeres a felhasználó egy gyakorlatra és el szeretné menteni magához, előbb id alapján meg kell keresni az adatbázisban. De ez egy Rateable objektum lesz, tehát melyik Repositoryban kell keresni? A felhasználó milyen objektumok közül választhat, hogy mihez adja hozzá?

Megoldás: egy RateableService interfész létrehozása, a Gym, Workout és Exercise egységes kezelésére. Attól függően, hogy milyen keresés történt adunk neki értéket. Például ha a felhasználó a gyakorlatok között keres akkor a RateableService egy ExerciseService értéket kap. Így már tudjuk, hogy a ExerciseRepositoryban kell az id-t keresni, illetve, hogy a felhasználó edzőterveit kell felsorolni, amikhez hozzáadhatja az adott gyakorlatot.

- Izomcsoport hozzáadása edzőtervhez

Edzőtervhez nem lehet izomcsoportot hozzáadni, hanem gyakorlat hozzáadásakor, a gyakorlat fő izomcsoportját, adja hozzá az edzőtervhez, ha még nem tartalmazza. Törléskor ugyanígy, gyakorlat törlésekor veszi ki a gyakorlat fő izomcsoportját, ha más gyakorlat nincs az edzőtervben, aminek ugyanaz a fő izomcsoportja.

- Adott állapot információk eltárolása

A TrackerApplication maga a program osztály, amely egy singleton. Ez alkalmas volt arra, hogy a jelenleg bejelentkezett felhasználót eltároljuk, amely null, ha például egy Visitor használja az alkalmazást, aki csak böngészni tud az adatbázisban. Szintén eltárolja a jelenleg kiválasztott edzőtermet, edzőtervet, vagy gyakorlatot (például: a felhasználó megnyitja egy edzőtermét, ahol kilistázza az edzőterveket, de melyik edzőterem edzőterv listáját írja ki). De eltárol egy Rateable objektumot is, amely a jelenleg kiválasztott keresési találat.

- Más felhasználó edzőtermének változtatása

A felhasználók, csak az általuk létrehozott edzőtermet, edzőtervet és gyakorlatot módosíthatják. Amikor a felhasználó egy publikus edzőtermet felvesz a sajátjai közé, amit nem ő hozott létre, és módosít rajta, akkor készül róla egy másolat az adatbázisba, más id-val, és annak már az a felhasználó a létrehozója. Ez nem lesz publikus, hogy a keresést ne szennyezze az eredetihez hasonló, de kicsit más edzőtermekkel.