

Edzési szokásokat nyilvántartó rendszer fejlesztése

Egy fitnesszterem hálózat digitális rendszert szeretne fejleszteni a vendégek edzési szokásainak nyilvántartására és elemzésére. A cél egy olyan backend alapú adatkezelő rendszer kialakítása, amely képes tárolni a vendégek edzéseinek adatait, valamint különböző statisztikákat és lekérdezéseket biztosítani a vezetőség számára.

A rendszer segítségével nyomon követhetővé válik:

- milyen típusú edzéseket végeznek a vendégek,
- milyen gyakran edzenek,
- mennyi időt töltenek sportolással,
- mely edzésformák a legnépszerűbbek,
- hogyan alakul az edzési aktivitás időszakonként.

A fejlesztési projekt backend csapatának feladata egy REST API elkészítése, amely:

- képes egy megadott adatállomány importálására adatbázisba,
- különböző szempontok szerinti lekérdezéseket biztosít,
- statisztikai kimutatásokat készít,
- támogatja az adatok módosítását meghatározott üzleti szabályok szerint,
- megfelelő hibakezelést alkalmaz.

A feladat célja egy jól strukturált, adatbázis-alapú backend szolgáltatás megvalósítása, amely támogatja a fitnesszterem működését és adatvezérelt döntéshozatalát.

Backend feladat – Edzési szokások API

Adatbázis létrehozása

Hozzon létre új adatbázist fitnessz néven!

Importálja a kapott edzesek.txt (UTF-8, tabulátorral tagolt) állományt az edzesek nevű táblába!

Az állomány első sora a mezőneveket tartalmazza.

Tábla szerkezete

edzesek

azon	Edzés azonosító (kulcs)	szám
nev	Vendég neve	szöveg
tipus	Edzés típusa (kardio, ero, joga, crossfit)	szöveg
ido	Edzés időtartama percben	szám
kaloria	Elégetett kalória	szám
datum	Edzés dátuma	dátum

1. GET /hosszu/:perc

Adja meg azoknak az edzéseknek az adatait (név, típus, idő), amelyek legalább a megadott percre tartottak!

Példa:

/hosszu/60

2. GET /vendeg

Query paraméterben kapott vendégnév alapján adja vissza az adott vendég összes edzését (típus, idő, dátum)!

Példa:

/vendeg?nev=Kiss Anna

3. GET /aktiv/:darab

Adja meg azokat a vendégeket, akik legalább a megadott számú edzésen vettek részt!

Példa:

/aktiv/5

4. PUT /modosit

Módosítsa egy adott edzés kalóriaértékét.

A kérés törzsében:

```
{  
  "azon": 12,  
  "kaloria": 650  
}
```

Szabályok:

Ha az edzés nem létezik → 404 hiba: "Nincs ilyen edzés"

Ha az új kalóriaérték kisebb, mint a jelenlegi → 400 hiba: "Nem megfelelő kalóriaérték"

Ha az új kalóriaérték kisebb mint 100 → 400 hiba: "Túl alacsony érték"

Egyéb esetben frissítse az adatot

5. POST /ujedzes

Rögzítsen új edzést az adatbázisba.

A kérés törzsében (JSON):

```
{  
  "nev": "Kiss Anna",  
  "tipus": "kardio",  
  "ido": 60,  
  "kaloria": 520,  
  "datum": "2025-03-10"  
}
```

Szabályok:

1. Kötelező mezők

Minden mező megadása kötelező.

Ha bármelyik hiányzik → 400 Bad Request

```
{ "message": "Hiányzó adatok." }
```

2. Érvényes edzéstípus

A típus csak az alábbi lehet:

kardio

ero

joga

crossfit

Ha ettől eltér → 400 Bad Request

```
{ "message": "Hibás edzéstípus." }
```

3. Időtartam validálása

Minimum: 20 perc

Maximum: 300 perc

Ha kívül esik a tartományon → 400 Bad Request

```
{ "message": "Érvénytelen időtartam." }
```

4. Kalória validálása

Minimum 100 kalória

Nem lehet nagyobb, mint az idő * 20

(pl. 60 perces edzésnél max 1200)

Ha nem felel meg → 400 Bad Request

```
{ "message": "Nem megfelelő kalóriaérték." }
```

5. Dátum ellenőrzés

Nem lehet jövőbeli dátum

Ha jövőbeli → 400 Bad Request

```
{ "message": "A dátum nem lehet jövőbeli." }
```

6. Sikeres rögzítés esetén

201 Created

```
{  
  "message": "Edzés sikeresen rögzítve",  
  "id": 25  
}
```