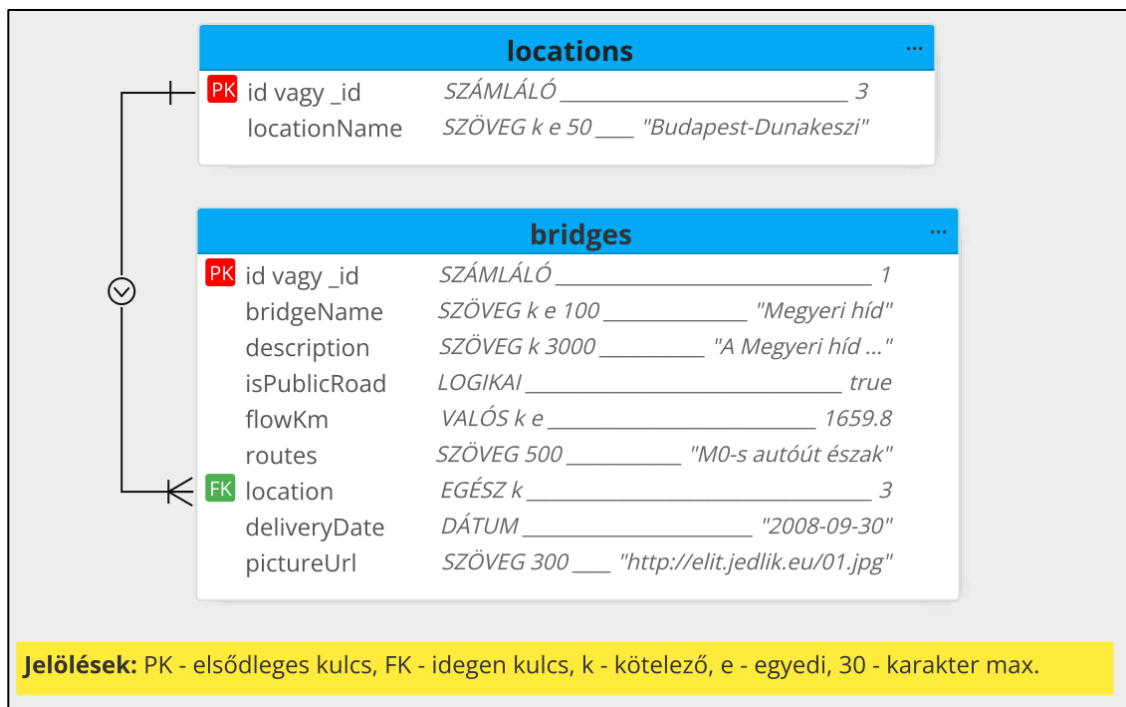


Backend programozás – Dolgozat - Magyarország Duna-hidjai

A következő feladatban Magyarország dunai hidjait bemutató WEB alkalmazáshoz kell backend szervert készítenie a leírás és a minták szerint! Megoldása csak úgy lesz teljes értékű, ha a projektjében a modell(ek) és kontroller(ek) pontosan a feladatspecifikáció szerintiek, felesleges kódot nem tartalmaznak.

1. Hozzon létre backend server projektet a választott technológia szerint! A projektmappát „Vezetéknév_Keresztnév_backend” formában nevezze el!
2. Hozzon létre modelleket a fenti ábra alapján a táblákhoz!



3. Végezze el a következő beállításokat a megfelelő modellben:

- Az `isPublicRoad` mező alapértelmezett értéke `true` legyen, ha az nem kerül megadásra!
- A `flowKm` mezőbe bevitt értékre teljesülni kell, hogy 2850-nél kisebb, viszont 0-nál nagyobb! Ha a feltétel nem teljesül, akkor „A `flowKm` mező értéke csak 0-nál nagyobb és 2850-nél kisebb lehet!” szöveg legyen megtalálható a hibaüzenetben!
- A `deliveryDate` mezőbe nem kerülhet az aktuális dátumnál nagyobb dátum! Ha a feltétel nem teljesül, akkor „Az aktuális dátumnál nem adhat meg későbbi dátumot a `deliveryDate` mezőben!” szöveg legyen megtalálható a hibaüzenetben!

4. Indítsa el számítógépén az adatbázis szervert, hozza létre az adatbázist `danubeBridges` néven! Töltse fel a `locations` táblát (kollekciót) az alábbi rekordokkal (dokumentumokkal):

id	locationName
1	Baja-Pörboly
2	Budapest
3	Budapest-Dunakeszi
4	Budapest-Szigetszentmiklós
5	Dunaföldvár-Solt
6	Dunaújváros-Dunavecse
7	Szekszárd-Dusnok

A végpontok teszteléséhez – a választott technológiától függően – a `bridges` táblához tartozó `bridges.csv` ill. `bridges.json` fájlban található minta adatokat használhatja.

5. Készítsen végpontot `/api/danube/bridges/:field/:direction` route-on, melynek használatával **GET** típusú kéréssel a `bridges` tábla összes adatát tudja a megadott mező (`field`) és rendezési irány (`direction` [ASC, DESC]) szerint rendezve lekérdezni! A dátum típusú mezőnél nem kötelező a ISO formátumú dátummal visszatérni! Oldja meg, hogy a `location` mezőbe ne a lokáció azonosítója (idegen kulcsa), hanem a kapcsolódó `locations` táblából kerüljön a válaszüzenetbe **objektumként** a lokáció neve (`locationName`) és azonosítója (`id` vagy `_id`) is! Egy lehetséges sikeres (200 OK) válasz részlete:

```
[
  {
    "_id": 13,
    "name": "Beszédes József híd",
    "description": "A Beszédes József ...",
    "isPublicRoad": true,
    "flowKm": 1560.56,
    "routes": "52-es főút",
    "location": {
      "_id": 5,
      "locationName": "Dunaföldvár-Solt"
    },
    "deliveryDate": "1930-11-23T00:00:00.000Z",
    "pictureUrl": "http://elit.jedlik.eu/nits/bridges/13.jpg"
  },
  ...
]
```

Method	Route	Body	Response
GET	<code>/api/danube/bridges/:field/:direction</code>	üres	JSON object array

6. Készítsen végpontot `/api/danube/bridges` route-al, melynek használatával **POST** típusú kéréssel új hidat tud rögzíteni az adatbázisban! A végponthoz tartozó kezelő (handler) kódsorait védje `try-catch` szerkezettel és hiba esetén a beépített hibaüzenetet küldje vissza a válaszüzenetbe a 400-as státuskóddal! Sikeres rögzítés után térjen vissza a 201 CREATED kóddal üzenet nélkül! Ügyeljen rá, hogy MongoDB server esetén az elsődleges kulcs (`_id`) értékét is meg kell adnia!

Egy lehetséges body:

```
{
  "_id": 123,
  "bridgeName": "Galvani-híd",
  "description": "A híd Budapest tervezett ...",
  "isPublicRoad": true,
  "flowKm": 1641,
  "routes": "Budapest, XI. és IX. kerület",
  "location": 2,
  "deliveryDate": "2024-11-30",
  "pictureUrl": "http://elit.jedlik.eu/nits/bridges/123.jpg"
}
```

Egy lehetséges válasz hiba (400 Bad Request) esetén:

"Az aktuális dátumnál nem adhat meg későbbi dátumot a `deliveryDate` mezőben!"

Method	Route	Body	Response
POST	<code>/api/danube/bridges</code>	JSON object	üres/szöveges

7. Készítsen végpontot a `/api/danube/bridges/:id` route-on, melynek használatával **PATCH** típusú kéréssel egy megadott azonosítójú híd tud módosítani az adatbázisban! Ha a megadott híd nem létezik, akkor „A híd `{id}` azonosítóval nem létezik!” üzenetet és a 404-es státuskódot küldje válaszként! A végponthoz tartozó kezelő (handler) kódsorait védje `try-catch` szerkezettel és hiba esetén a beépített hibaüzenetet küldje vissza a válaszüzenetbe a 400-as státuskóddal! Sikeres módosítás esetén a 200-as státuskódot és a módosított rekordot (dokumentumot) küldje válaszként!

Minta body:

```
{
  "routes": "M0-s autóút északi része",
}
```

Minta response sikeres módosítás után:

```
{
  "_id": 1,
  "bridgeName": "Megyeri híd",
  "description": "A Megyeri híd...",
  "isPublicRoad": true,
  "flowKm": 1659.8,
  "routes": "M0-s autóút északi része",
  "location": 3,
  "deliveryDate": "2008-09-30T00:00:00.000Z",
  "pictureUrl": "http://elit.jedlik.eu/nits/bridges/01.jpg"
}
```

Minta hibaüzenetek:

"A híd 123 azonosítóval nem létezik!"

"Az aktuális dátumnál nem adhat meg későbbi dátumot a deliveryDate mezőben!"

Method	Route	Body	Response
PATCH	<code>/api/danube/bridges/:id</code>	JSON object	JSON object/szöveges

8. Hozzon létre **Thunder Client** kollekciót “Backend teszt” néven melyben az Ön által létrehozott végpontokat ellenőrzi! Exportálja a kollekciót a projektmappába “Backend teszt.json” néven!
9. **Projektmappáját** tömörítve másolja a “Z:” megható főkönyvtárába! (Vezetéknév_Keresztnév_backend.ZIP néven) Tömörítés előtt törölje a felesleges állományokat! **Ügyeljen arra, hogy feladatmegoldást tartalmazó mappát/állományt ne töröljön!**