

Backend programozás – Északi-középhegység kilátói

Ebben a feladatban az Északi-középhegység néhány kilátójának fontosabb adatait kiszolgáló backend-szerver alkalmazást kell elkészítenie a leírás és a minták alapján.

Megoldása csak akkor lesz teljes értékű, ha a projektjében a modell(ek) és kontroller(ek) pontosan a feladatspecifikáció szerintiek, felesleges kódot nem tartalmaznak.

1. Hozzon létre backend-szerver projektet az Ön által választott programnyelven! A projektmappát „Vezetéknév_Keresztnév_backend” formában nevezze el!
2. Indítsa el a választott lokális adatbázis szervert, majd futtassa a Forrás mappában lévő mongo.bat állományt az adatbázis (ncmviewpoints) és a táblák (kollekciók) létrehozásához, valamint a mintaadatok feltöltéséhez.
3. Hozzon létre modelleket az alábbi ábra alapján! A locations modell a hegységek, a viewpoints modell pedig a kilátókhoz tartozó tulajdonságokat határozzák meg.
A modellekben állítsa be a mezőket, adattípusokat és a jelzett megszorításokat!

locations	típus és megszorítás	példa
_id locationName	Egész szám, kötelező, elsődleges kulcs Szöveg, egyedi, kötelező, max: 30 karakter	1 "Börzsöny"
viewpoints	típus és megszorítás	példa
_id viewpointName mountain height description built imageUrl location_id	Egész szám, kötelező, elsődleges kulcs Szöveg, egyedi, kötelező, max: 50 karakter Szöveg, egyedi, kötelező, max: 30 karakter Tizedes törtet tartalmazó szám (double) Szöveg, kötelező Dátum Szöveg, kötelező, max: 50 karakter Egész szám, kötelező, idegen kulcs	2 "Julianus-kilátó" "Hegyes-tető" 15,2 "A csaknem tizen..." 1939-05-14 ".../julianus.jpg" 1

4. Továbbá állítsa be a következő megszorításokat is a megfelelő modellben:
 - a. Ha az imageUrl mező értéke nem kerül megadásra, akkor az alapértelmezett érték: "<https://nits68.github.io/static/viewpoints/no-img.jpg>" legyen!
 - b. A built mezőbe nem kerülhet az aktuális dátumnál nagyobb dátum! Ha a feltétel nem teljesül, akkor "Az aktuális dátumnál nem adhat meg későbbi dátumot a built mezőben!" szöveg legyen megtalálható a hibaüzenetben!
 - c. A height mezőbe legalább 1 m magasságot kell megadni a kilátó magasságának! Ha a feltétel nem teljesül, akkor "Egy kilátónak legalább 1 méter magasnak kell lennie!" szöveg legyen megtalálható a hibaüzenetben!

A feladatok megoldása során ügyeljen arra, hogy a megoldása az elvárt végpontokon kívül egyéb végpontokat ne tartalmazzon! A végpontok funkcionálisát megvalósító kontrollerekben a szükséges metódusokon kívül felesleges kódokat ne hozzon létre!

5. Hozzon létre egy végpontot, amely egy megadott nevű hegység (`locationName`) kilátóit listázza ki! Amennyiben van legalább egy találat, akkor adja vissza a talált kilátókat. Ha egyetlen kilátó sincs az adott hegységen, vagy a hegység nem létezik az adatbázisban, akkor a **minták szerinti** válaszüzenetet küldje vissza!

Metódus	URL	Body	Válasz
GET	/api/locations/:locationName/viewpoints (pl.: <code>localhost:3000/api/locations/Bükk/viewpoints</code>)	üres	JSON array/ JSON object

Például (**200 OK** státuszkóddal):

```
[  
  {  
    "viewpointName": "Kalapat-hegyi kilátó",  
    "mountain": "Kalapat-tető",  
    "location_id": 2,  
    "height": 33.3,  
    "description": "A kilátótoronyból körpanoráma...",  
    "built": "2000-08-20T00:00:00.000Z",  
    "imageUrl": "https://nits68.github.io/.../kalapat.jpg"  
  }  
  ...  
]
```

Minta válaszüzenetek (**404 NOT FOUND** státuszkóddal):

```
{"message": "Ebben a hegységen nem találtam kilátót!"}
```

6. Készítsen végpontot a /api/viewpoints/:id route-on, melynek használatával **PATCH** típusú kéréssel a megadott azonosítójú kilátó adatait tudja módosítani az adatbázisban! Ha a megadott kilátó nem létezik, akkor a "Ezzel az azonosítóval ({id}) nem létezik kilátó!" üzenetet és a 404-es státuszkódot küldje vissza válaszként! A végponthoz tartozó kódsorokat védje try-catch szerkezettel és hiba esetén a beépített hibaüzenetet küldje vissza a válaszüzenetben 400-as státuszkóddal! Sikeres módosítás esetén a 200-as státuszkódot és a módosított dokumentumot (rekordot) küldje válaszként!

Metódus	URL	Body	Válasz
PATCH/PUT	/api/viewpoints/:id	JSON object	JSON object
Minta body (PATCH):	Minta body (PUT):		
{ "height": 30.0, "built": "2015-05-04" }	{ "viewpointName": "Galya-kilátó", "mountain": "Galya-tető", "location_id": 4, "height": 30, "description": "Az ország legmagasabban álló...", "built": "2015-05-04T00:00:00.000Z", "imageUrl": " https://nits68.github.io/.../galya.jpg " }		

Minta válasz sikeres módosítás után:

```
{
  "_id": 10,
  "viewpointName": "Galya-kilátó",
  "mountain": "Galya-tető",
  "location_id": 4,
  "height": 30,
  "description": "Az ország legmagasabban álló...",
  "built": "2015-05-04T00:00:00.000Z",
  "imageUrl": "https://nits68.github.io/.../galya.jpg"
}
```

Minta hibaüzenetek:

```
{"message": "19 azonosítóval nem létezik kilátó!"}

{"message": "Validation failed: built: Az aktuális dátumnál nem adhat meg későbbi dátumot a built mezőben!"}
```

7. Hozzon létre Postman kollekciót, melyben az Ön által létrehozott végpontokat ellenőrzi! Exportálja a kollekciót a projekt gyökérkönyvtárába teszt.json néven!

8. Adja le a teljes projektmappát tömörítve (node_modules mappa nélkül)!