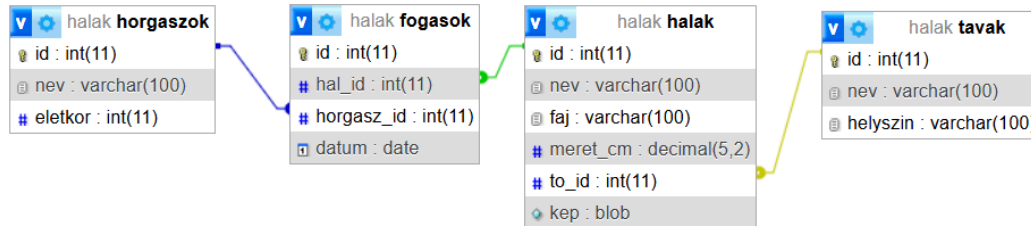


HalakAPI

1. Importálja a csatolt halak.sql állományt az adatbáziskezelőbe (XAMPP).
A következő szerkezetű adatbázist kell kapnia:
2. Készítsen HalakAPI néven ASP.Net Core Web Api alkalmaz



ást.

3. Adja a projekthez (.NET 8.0) a következő csomagokat (ügyeljen a helyes verziószámokra!):
 - a. Microsoft.EntityFrameworkCore 8.0.15
 - b. Microsoft.EntityFrameworkCore.Tools 8.0.15
 - c. MySQL.EntityFrameworkCore 8.0.14
4. A megfelelő csomagok telepítése után adja ki a következő parancsot:
Scaffold-DbContext
"SERVER=localhost;PORT=3306;DATABASE=halak;USER=root;PASSWORD=;SSL
MODE=none;" mysql.entityframeworkcore -outputdir Models -f
5. Helyezze el a körkörös Json hivatkozás megszüntetéséhez szükséges kiegészítéseket a létrejött osztályokban, illetve gondoskodjon arról, hogy a megfelelő virtuális property-k null értékűek lehessenek!
vagy ezt az utasítást illeszk be a megfelelő helyre:
`builder.Services.AddControllers().AddJsonOptions(x =>
x.JsonSerializerOptions.ReferenceHandler = ReferenceHandler.IgnoreCycles);`
6. Alakítson ki végpontokat a következőkhöz:
 - a. A horgászok minden adatát visszadó végpont.

GET /Horgaszok/All

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	A horgászok adataiból álló lista
Hiba a feldolgozás közben	400	A hiba okára utaló üzenet

b. Adott azonosítójú horgász adatait visszaadó végpont

GET /Horgaszok/ById/{id} ▾

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	A horgász adataiból álló objektum
Hiba a feldolgozás közben	400	A hiba okára utaló üzenet
Hiányzó id	404	"Nincs ilyen azonosítójú horgász!"

- c. Készítsen a halak adataihoz egy osztályt, amiben csak a halak faja, mérete és ha van, akkor a tó neve szerepel, ahol kifogták.
- d. A c. pontban elkészített osztályt felhasználva adja meg a kifogott halak listáját, benne a fajtát, méretét és a tó nevét .

GET /Halak/FajMeretTo ▾

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	A halak adataiból álló lista
Hiba a feldolgozás közben	400	A hiba okára utaló üzenet

e.

Hozzon létre végpontot új hal rögzítésére az alábbi feltételekkel és beállításokkal! A hal rögzítése csak akkor lehetséges, ha a felhasználó be van jelentkezve és van jogosultsága hozzá. Ez akkor lehetséges, ha a paraméterben kapott „UserID” értéke „FKB3F4FEA09CE43C”. Oldja meg, hogy a főprogram tárolja ezt az azonosítót „UID” néven és a végpont hasonlítsa össze ezt a paraméterében kapott értékkel. Sikeres rögzítés esetén 200 OK státuszkóddal és a „Hal hozzáadása sikeresen megtörtént.” JSON üzenettel térjen vissza! Hiba esetén 400 BAD REQUEST hibakóddal és a hibára utaló üzenettel térjen vissza! Amennyiben jogosultsági probléma lép fel, akkor 401 Unauthorized hibakóddal és „Nincs jogosultsága új hal felvételéhez!” üzenettel térjen vissza.

POST /Halak ▾

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	"Hal hozzáadása sikeresen megtörtént."

Hiba a feldolgozás közben	400	A hiba okára utaló üzenet
Nincs jogosultsága	401	"Nincs jogosultsága új hal felvételéhez!"

- f. Készítsen egy hal adatainak módosítására szolgáló aszinkron végpontot, ami paraméterként egy hal objektumot kap a kérés törzsében.

PUT /Halak

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	"Sikeres módosítás."
Hiba a feldolgozás közben	400	A hiba okára utaló üzenet
Hiányzó azonosító esetén	404	"Nincs ilyen azonosítójú hal!"

- g. Készítsen egy új hal adatainak törlésére szolgáló aszinkron végpontot, ami paraméterként kapja a törlendő hal azonosítóját.

DELETE /Halak

Eset	Státuszkód	Válasz tartalma
Hibátlan futás	200	"Sikeres törlés."
Hiba a feldolgozás közben	400	A hiba okára utaló üzenet
Hiányzó azonosító esetén	404	"Nincs ilyen azonosítójú hal!"

- h. A „CORS” műveletek beállításához illessze be a következő két programrészletet a megfelelő helyekre.

a) `builder.Services.AddCors(c => { c.AddPolicy("AllowOrigin", options => options.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader()); });`

b) `app.UseCors(options => options.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader());`