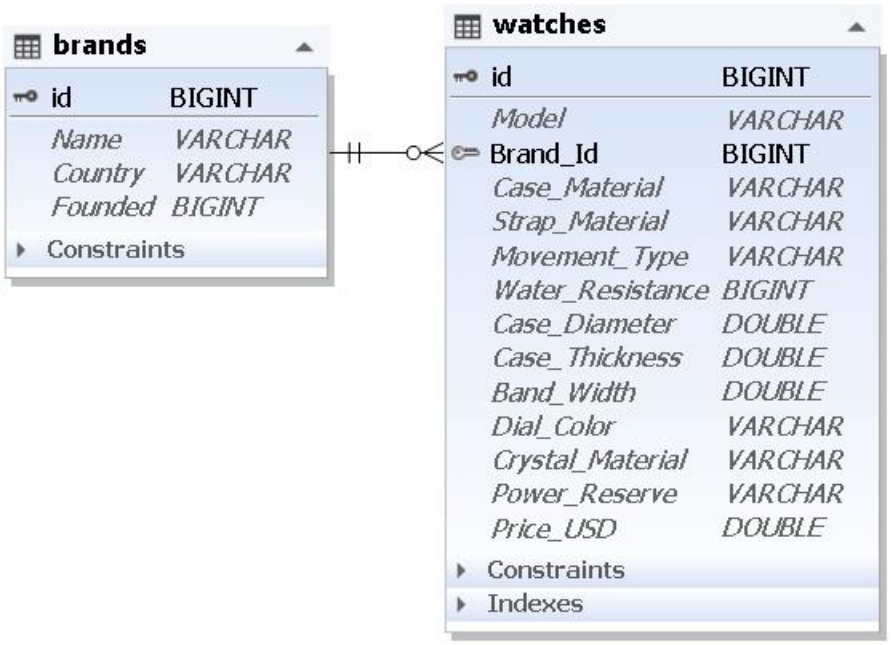
Laravel backend programozás – Luxus órák feladat

Az alábbi feladatban Önnek luxus kategóriába tartozó órák nyilvántartását végző backend alkalmazást kell elkészítenie.

1. Hozzon létre Laravel projektet!
   A projektet „`Vezetéknév_Keresztnév_luxury_watches_api`” formában nevezze el!

2. Hozzon létre adatbázist a `luxury_watches.sql` állomány lefuttatásával!

3. A lenti ábra alapján hozza létre a szükséges modell és controller osztályokat és metódusokat!



4. Hozzon létre végpontot az adatbázisban szerepelő **összes óra** lekérdezésére az alábbi beállításokkal! A válasz **200 OK** státuszkód mellett az órák adatait tartalmazó tömb legyen! Gondoskodjon róla, hogy a visszaadott adatok között ne csak a Brand_id hanem a Barand többi adatai is szerepeljen!

Metódus	URL	Body	Válasz
GET	/api/watches	üres	JSON tömb

5. Hozzon létre végpontot az adatbázisban szerepelő **egy óra** lekérdezésére az alábbi beállításokkal! A `{watch}` helyén a keresett óra azonosítója (`id`) szereplejen! Létező `id` esetén **200 OK** státuszkóddal és a keresett óra adataival térjen vissza! Amennyiben a megadott `id` nem létezik **404 NOT FOUND** státuszkóddal térjen vissza!

Metódus	URL	Body	Válasz
GET	/api/watches/{watch}	üres	JSON objektum

6. Hozzon létre végpontot **új óra rögzítésére** az alábbi beállításokkal! Sikeres rögzítés esetén **201 CREATED** státuszkóddal és az újonnan beszúrt rekord adatait tartalmazó JSON üzenettel térjen vissza! Gondoskodjon az adatok validálásáról! Bármely kötelező mező hiánya vagy nem megfelelő típusa esetén **422 UNPROCESSABLE CONTENT** hibakóddal térjen vissza! A `Brand_id`-ban csak olyan értéket fogadjon el, amely létezik a brands táblában!

Metódus	URL	Body	Válasz
POST	/api/watches	JSON	JSON objektum

Példa a kérés body-ban elküldött JSON tartalmára:

```
{
  "Model": "Royal oak",
  "Brand_Id": 2,
  "Movement_Type": "Automatic",
  "Water_Resistance": 200,
  "Price_USD": 3.3
}
```

7. Hozzon létre végpontot az adatbázisban szerepelő **óra törlésére** az alábbi beállításokkal! A `{watch}` helyén a törölni kívánt óra azonosítója szereplejen! Sikeres törlés esetén **204 NO CONTENT** státuszkóddal térjen vissza! Amennyiben a megadott `id` nem létezik **404 NOT FOUND** státuszkóddal térjen vissza!

Metódus	URL	Body	Válasz
DELETE	/api/ watches/{watch}	üres	üres/JSON

8. Hozzon létre végpontot **meglévő óra módosítására** az alábbi beállításokkal! A `{watch}` helyén a módostani kívánt óra azonosítója szereplejen! Sikeres módosítás esetén **200 OK** státuszkóddal és a korábbi feladatokhoz hasonlóan a módosított rekord adatait tartalmazó JSON üzenettel térjen vissza! Amennyiben a megadott `id` nem létezik **404 NOT FOUND** státuszkóddal térjen vissza!

Metódus	URL	Body	Válasz
PATCH	/api/watches/{watch}	JSON	JSON

Példa a kérés body-ban elküldött JSON tartalmára:

```
{
  "Price_USD": 5.5
}
```

9. Hozzon létre végpontot egy adott **márka (brand) óráinak**nak megjelenítésére! A `{brand}` helyén a kívánt márka azonosítója szereplejen! ! Létező `id` esetén **200 OK** státuszkóddal és a keresett márkához tartozó órák adataival térjen vissza! Amennyiben a megadott azonosító nem létezik **404 NOT FOUND** státuszkóddal és a rekord hiányára utaló üzenettel térjen vissza!

Metódus	URL	Body	Válasz
GET	/api/brands/{brand}/watches	üres	JSON tömb/JSON objektum

10. Hozzon létre Thunder Client kollekciót "`Backend teszt`" néven, melyben az Ön által létrehozott összes útvonalat ellenőrzi! Exportálja a kollekciót a projekt gyökér könyvtárába "`Backend_teszt`" néven!

11. A projekt leadása

    a Törölje a `vendor` mappát!
    b Tömörítse be a projekt mappáját `Vezetéknév_Keresztnév_backend.zip` néven!
    c Másolja a tömörített állományt a megadott helyre!