

Laravel backend programozás – Nyomdai megrendelések feladat

Az alábbi feladatban Önnek egy nyomda megrendeléseinek nyilvántartását végző backend alkalmazást, és a hozzá tartozó adatbázist kell elkészítenie.

1. Hozzon létre backend szerver projektet!
A projektet „Vezetéknév_Keresztnév_press” formában nevezze el!
2. Készítsen adatbázist `press` néven!
3. A lenti ábra alapján hozzon létre modelleket, migrációkat és egyéb szükséges osztályokat, majd definiálja a megfelelő adattípusokat a szükséges beállításokkal!



4. Hozzon létre `OrderFactory` néven osztályt, amelynek segítségével fel tudja tölteni az `orders` táblát a mezők típusának leginkább megfelelő generált (fake) adatokkal!
 5. A `DatabaseSeeder` osztály felhasználásával:
 - a. töltse fel a `categories` táblát az alábbi rekordokkal:
- | id | type |
|----|----------|
| 1 | könyv |
| 2 | magazin |
| 3 | napilap |
| 4 | szórólap |
- b. az `OrderFactory` osztály példányosításával töltse fel legalább 20 rekorddal az `orders` táblát.
6. Futtassa le a migrációkat és töltse fel a táblákat adatokkal a `DatabaseSeeder` osztály segítségével!
 7. Hozzon létre végpontot az adatbázisban szereplő **összes megrendelés** lekérdezésére az alábbi beállításokkal! Oldja meg, hogy a válasz üzenetben a **kiadvány típusa** is szerepeljen!

| Metódus | URL | Body | Válasz |
|------------|-------------|------|--------|
| GET | /api/orders | üres | JSON |

Lehetséges válasz üzenet (**200 OK** státuszkóddal):

```
[
  {
    "id": 1,
    "category": {
      "id": 3,
      "type": "napilap"
    },
    "customer_name": "Adam Haley",
    "deadline": "1980-10-06",
    "quantity": 560,
    "publisher_name": "Justine Adams"
  },
  ...
]
```

8. Hozzon létre végpontot az adatbázisban szereplő **egy megrendelés** lekérdezésére az alábbi beállításokkal! Az {id} helyén a keresett megrendelés azonosítója szereplejen! Létező id esetén **200 OK** státuszkóddal és a keresett megrendelés adataival térjen vissza! Amennyiben a megadott id nem létezik **404 NOT FOUND** státuszkóddal és a „Megrendelés nem létezik” JSON üzenettel térjen vissza!

| Metódus | URL | Body | Válasz |
|------------|------------------|------|--------|
| GET | /api/orders/{id} | üres | JSON |

Lehetséges válasz üzenet (**200 OK** státuszkóddal):

```
[
  {
    "id": 1,
    "category": {
      "id": 3,
      "type": "napilap"
    },
    "customer_name": "Adam Haley",
    "deadline": "1980-10-06",
    "quantity": 560,
    "publisher_name": "Justine Adams"
  }
]
```

Válasz üzenet (**404 NOT FOUND** státuszkóddal):

```
{
  "Message": "A megrendelés nem létezik."
}
```

9. Hozzon létre végpontot **új megrendelés rögzítésére** az alábbi beállításokkal! Sikeres rögzítés esetén **201 CREATED** státuszkóddal és az újonnan beszúrt rekord azonosítóját tartalmazó JSON üzenettel térjen vissza! Bármely kötelező mező hiánya esetén **400 BAD REQUEST** hibakóddal és a hibára utaló „Hiányos adatok” JSON üzenettel térjen vissza!

| Metódus | URL | Body | Válasz |
|-------------|-------------|------|--------|
| POST | /api/orders | JSON | JSON |

Példa a kérés body-ban elküldött JSON tartalmára:

```
{
  "category": 3,
  "customer_name": "John Doe",
  "deadline": "2022-03-22",
  "quantity": 10,
  "publisher_name": "Jane Doe"
}
```

Válasz üzenet (**201 CREATED** státuszkóddal):

```
{
  "id": 21
}
```

Válasz üzenet (**400 BAD REQUEST** státuszkóddal):

```
{
  "Message": "Hiányos adatok."
}
```

10. Hozzon létre végpontot az adatbázisban szerepelő **megrendelés törlésére** az alábbi beállításokkal! Az {id} helyén a törölni kívánt megrendelés azonosítója szereplejen! Sikeres törlés esetén **204 NO CONTENT** státuszkóddal térjen vissza! Amennyiben a megadott id nem létezik **404 NOT FOUND** státuszkóddal és a „Megrendelés nem létezik” JSON üzenettel térjen vissza!

| Metódus | URL | Body | Válasz |
|---------------|------------------|------|-----------|
| DELETE | /api/orders/{id} | üres | üres/JSON |

Válasz üzenet (**404 NOT FOUND** státuszkóddal):

```
{
  "Message": "A megrendelés nem létezik."
}
```

11. Hozzon létre végpontot **meglévő megrendelés módosítására** az alábbi beállításokkal! Az {id} helyén a módosítani kívánt megrendelés azonosítója szereplejen! Sikeres módosítás esetén **200 OK** státuszkóddal és a módosított rekord adatait tartalmazó JSON üzenettel térjen vissza!

| Metódus | URL | Body | Válasz |
|------------|------------------|------|--------|
| PUT | /api/orders/{id} | JSON | JSON |

Példa a kérés body-ban elküldött JSON tartalmára:

```
{
  "quantity": 30
}
```

Válasz üzenet (**200 OK** státuszkóddal):

```
{
  "id": 21,
  "category": 3,
  "customer_name": "John Doe",
  "deadline": "2022-03-22",
  "quantity": 30,
  "publisher_name": "Jane Doe"
}
```

Válasz üzenet (**400 BAD REQUEST** státuszkóddal):

```
{
  "Message": "Hiányos adatok."
}
```

12. 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!
13. Adja le:
 - a. a program forráskódját tömörítve
 - b. az adatbázist létrehozó SQL scriptet