Backend építése Laravelben

## Összes eddigi tananyag felfűzve [itt](https://docs.google.com/document/d/1ybnWk0tEwyhRDPbPCpyBNf4YEybYqK1jbMC2fahuEJ4/edit?usp=sharing) található (folyamatosan bővül).

A múlt heti tananyag már beépítve található [itt](https://github.com/PIKSSL/book_with_multiply_keys.git) (többszörös primary key megoldva!), használd bátran kiindulásként a mai órához!

Az importálandó json fájlt megtalálod a json mappában a fenti linken.

* Vegyél fel 2 újabb mezőt a reservations táblában *message\_date* és *status* néven!
* Előbbi az üzenet dátumát jelzi, utóbbi a státuszát (alapból látható: 0, rejtett: 1)

Hozz létre még egy réteget (admin: *admin*, librarian: *könyvtáros,* basic - *kölcsönző*) a [korábbi tananyag](https://docs.google.com/document/d/1iw5NeT1m4c3oe6U0sNr1HL_yp5VRrrO1Hh5476NKmtc/edit?usp=sharing) alapján, és válaszd szét az eddigi végpontokat aszerint, hogy kinek mit kellene látnia/módosítani szerinted!

*A felhasználó* szeretne tudni pl.:

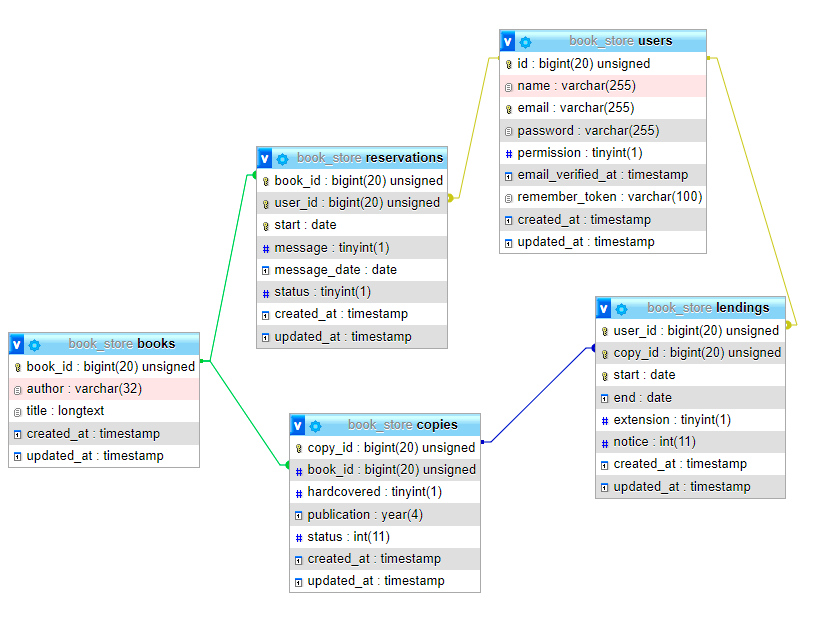
* keresni (listázni) könyvek között (pl. van-e a könyvtárban egy bizonyos könyv - nem selejtes)
* előjegyezni azon könyveket, amik nincsenek a raktárban (ha az előző lista count-ja 0, megjelenhet frontend-en egy gomb pl.)
* törölni az előjegyzést (status: 1),
* kikölcsönözni egy könyvet (post a lendings táblában) azon könyvek közül, amik elérhetőek, (copies táblában a statust itt át kell írni)
* saját felhasználói adatait kezelni

*A könyvtáros* szeretne tudni pl.:

* kilistázni bármelyik felhasználó
  + kölcsönzéseit
  + előjegyzéseit
  + adatait
* példányok státuszát esetlegesen 2-re állítani (leselejtezni),
* felvinni adatokat egy-egy példányról (post a copies táblára)

*Az admin* szeretne tudni pl.:

* törölni selejtezésre ítélt példányokat,
* törölni rejtettre állított előjegyzéseket,
* törölni felhasználókat, akik régóta nem aktívak…



## 

## Lekérdezések Laravelben[[1]](#footnote-0)

1. Eloquant használatával (**with** segítségével)
2. DB::table …
   1. **select (->get()),**
   2. **where**, whereNot, orWhere, whereNull, whereIn, whereDate, whereColumn…
   3. **aggregáló függvények (pl. ->count()),**
   4. **groupBy, having**, orderBy
   5. **raw**, azaz nyers sql használata
   6. **join**-ok használata
   7. unió,
   8. **view[[2]](#footnote-1)**-k sql-ben,
   9. triggerek,
   10. **constraintek**

## Lekérdezések DB::table segítségével

importálás: use Illuminate\Support\Facades\DB;

Alap szintaxis:

// egy adott könyvhöz hány példány tartozik

public function book\_copy\_count($title){

$copies = DB::table('copies as c') //egy tábla lehet csak

//->select('mezo\_neve') //itt nem szükséges

->join('books as b' ,'c.book\_id','=','b.book\_id') //kapcsolat leírása, akár több join is lehet

->where('b.title','=', $title) //esetleges szűrés

->count(); //esetleges aggregálás; ha select, akkor get() a vége

return $copies;

}

## 

## Konkrét lekérdezések az adatbázisban (DB::table gyakorlása)

**Elsőként írjuk meg a fent megírt lekérdezéseket ezzel a módszerrel is!**

1. Hány darab előjegyzése van a bejelentkezett felhasználónak?
2. Csoportosítsd szerzőnként a könyveket (nem példányokat) a szerzők ABC szerinti növekvő sorrendjében!
3. Határozd meg a könyvtár nyilvántartásában legalább 2 könyvvel rendelkező szerzőket!
4. A B betűvel kezdődő szerzőket add meg!
5. **A bejelentkezett felhasználó 3 napnál régebbi előjegyzéseit add meg! (együtt)**
6. **Bejelentkezett felhasználó azon kölcsönzéseit add meg (copy\_id és db), ahol egy példányt legalább db-szor (paraméteres fg) kölcsönzött ki! (együtt)**
7. **Hosszabbítsd meg a könyvet, ha nincs rá előjegyzés! (együtt)**

**Innentől önállóan:**

1. Kölcsönözz ki egy könyvet (post a lendings táblában) azon könyvek közül, amik elérhetőek, (egyúttal a copies táblában a statust át kell írni 1-re!)
2. Rejtsd el azokat a rekordokat a reservations táblában, amikre jött üzenet, és 10 napnál régebbiek (status: 1)
3. Listázd ki a mai napon visszahozott könyveket!
4. Listázd ki azon könyveket, amikre előjegyzést vettek fel, és még nincs értesítésük, és nincs rejtve!
5. Értesítsd azon felhasználókat, akiknek visszahozták azáltala előjegyzett könyv egy példányát, de ha ugyanarra a könyvre több előjegyzés is van, akkor a legrégebben előjegyző felhasználó kapjon üzenetet (és egyúttal tedd félre (status: 3) neki az adott könyvet).

A lekérdezésekhez létrehozott útvonalat teszteld TC-ben is, a böngészőben is (mind GET kérés)!

## Validációs feladat

Tegyük fel, hogy csak 1000 vagy azután kiadott könyvek találhatóak a könyvtárban.

Csak akkor engedd módosítani a copies táblában a publication nevű mezőt, ha 1-sel vagy 2-sel kezdődik az év, és 4 jegyű a szám!

(ehhez nem lesz jó a year típus, mert nem engedi a régebbi éveket, helyette:  
$table->integer('publication')->length(4)->default(2000);)

**Készíts legalább 5 működő Unit tesztet a meglévők alapján a lekérdezésekhez!**

1. <https://laravel.com/docs/9.x/queries> [↑](#footnote-ref-0)
2. <https://www.nicesnippets.com/blog/how-to-use-mysql-view-in-laravel-9> [↑](#footnote-ref-1)