FACTORY

Ha most kezdjük az elejéről:

//cmf: controllerrel, migr f-lal, factoryvel

*php artisan make:model Task -cmf*

Ha már van model, és controllert is szeretnénk:

*php artisan make:controller TaskController -m*

API függvényekkel létrehoz kontrollert:

*php artisan make:controller PhotoController --api*

Factory létrehozása:

*php artisan make:factory UserFactory --model=User*

kiegészítések:

magyar nyelvű adatok: *fake('hu\_HU')*  
válogatunk meglévő adatokból - Factory.php:  
*return [*

*'user\_id' => User::all()->random()->id,*

*'book\_id' => Book::all()->random()->book\_id,*

*];*

futtatása:

*php artisan db:seed*

migrálással:

*php artisan migrate:fresh --seed*

db>app>http/controllers>Lending.php

//alapfüggvények amiket minden controllerben használunk

public function index() {

return Lending::all(); //select \* from Lendings}

//egy rekord létrehozása

//modellnek megfelelő példány kitöltése az összes kérésnek megfelelőenˇ

public function store(Request $request) {

$record = new Lending();

$record->fill($request->all()); // létrehozunk egyúj rekordot és utána kitöltjük az összeset

$record->save(); //elmentjük a módosításokat

}

public function show(string $user\_id, $copy\_id, $start) {

$lending = Lending::where('user\_id', $user\_id) //paramétereket megnézi, hopgy megfelelnek e a különböző mezőknek

->where('copy\_id', $copy\_id)

->where('start', $start)

->get();

return $lending[0]; //lekérdezi, hogy létezik e ilyen rekord, visszatér egy listával, aminek az egyetlen eleme lesz az a rekord

public function update(Request $request, $user\_id, $copy\_id, $start) //ahhoz hogy módosítani tudjunk valamit meg kell adni paraméterekkel, hogy mit módosítanánk

{

$record = $this->show($user\_id, $copy\_id, $start);

$record->fill($request->all()); //azt a rekordot átírja?

$record->save();

}

public function destroy($user\_id, $copy\_id, $start{

$this->show($user\_id, $copy\_id, $start)->delete(); //kitörli a megadott paraméterekkel rendelkező rekordot

}

}

db>app>Models>Lending.php

class Lending extends Model{

use HasFactory;

protected $fillable =[

'user\_id',

'copy\_id',

'start'];

protected function setKeysForSaveQuery($query) { //KÜLSŐ KULCSOK

$query

->where('user\_id', '=', $this->getAttribute('user\_id'))

->where('copy\_id', '=', $this->getAttribute('copy\_id'))

->where('start', '=', $this->getAttribute('start'));

return $query;

}

}

db>database>factories>LendingFactori.php

class LendingFactory extends Factory{

public function definition(): array{

return [

'user\_id' => User::all()->random()->id,

'copy\_id' => Copy::all()->random()->copy\_id,

'start' => fake()->date() ];}}

//létrehozza a sorokat random adatokkal, de a külső kulcsoknál a másik tábla kulcsát írja

/\*UserFactoryban pl.:

'name' => fake('hu\_HU')->name(),

'email' => fake()->unique()->safeEmail(),

'email\_verified\_at' => now(),\*/

db>database>migrations>creat lendings

return new class extends Migration{

public function up(): void {

Schema::create('lendings', function (Blueprint $table) {

/\* $table->id(); \*/

$table->primary(['user\_id', 'copy\_id','start']);

$table->foreignId('user\_id')->references('id')->on('users');

$table->foreignId('copy\_id')->references('copy\_id')->on('copies');

$table->date('start')->default('2020-01-01');

$table->timestamps();

});

}

public function down(): void

{

Schema::dropIfExists('lendings');

}

};

DatabaseSeeders.php

public function run(): void

{

User::factory(10)->create();

Book::factory(10)->create(); //ez hozza létre a random sorokat a táblákba

Copy::factory(10)->create();

Lending::factory(10)->create();

}

db>routes

Route::get('/user', function (Request $request) {

return $request->user();

})->middleware('auth:sanctum');

//minden tábla adatait hasonlóképpen tudjuk elérni/kezelni-> át kell írni a táblaneveket!

//LENDING

Route::get('/lendings', [LendingController::class, 'index']);

Route::post('/lending', [LendingController::class, 'store']);

Route::get('/lending/{user\_id}/{copy\_id}/{start}', [LendingController::class, 'show']);

Route::put('/lending/{user\_id}/{copy\_id}/{start}', [LendingController::class, 'update']);

Route::delete('/lending/{user\_id}/{copy\_id}/{start}', [LendingController::class, 'destroy']);

Thunder client-be mennek a testek (get,post,put,delete)

Az urlbe mehet a localhost:8000/api/

Létrehozzuk a requestek és elküldjük őket.