### Eloquent

Laravel uključuje Eloquent, objektno-relacioni maper (ORM) koji čini interakciju sa vašom bazom podataka prijatnom. Kada koristite Elokuent, svaka tabela baze podataka ima odgovarajući „Model“ koji se koristi za interakciju sa tom tabelom. Pored preuzimanja zapisa iz tabele baze podataka, Elokuent modeli vam omogućavaju da ubacite, ažurirate i izbrišete zapise iz tabele.

Model možemo kreirati pomoću naredbe:php artisan make:model Course

, a obrisati ručno na lokaciji app.

Opciono: php artisan make:model Course --table=kurs

Laravel će automatski generisati model sa osnovnom strukturom. Modeli u Laravelu se obično nalaze u direktorijumu app i imaju .php ekstenziju.

Nakon što model bude kreiran, možeš ga dalje prilagoditi dodavanjem relacija, validacija i drugih funkcionalnosti prema potrebama tvoje aplikacije.

Uz model možemo da kreiramo mograciju:php artisan make:model Course --migration

iliphp artisan make:model Course –m

ili samo migracijuphp artisan make:migration create\_course\_table --table=course

Možemo da kreiramo i controler, migraciju i factory: php artisan make:model Course –mfc

Primjer migracije za course:

public function up(): void

{

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

        $table->id();

        $table->string('naziv');

        $table->enum('status', ['aktivan', 'neaktivan'])->default('aktivan');

        $table->integer('ECTS')->unsigned();

        $table->timestamps();

    });

}

Da bi migracija bila vidljiva na bazi potrebno je izvršiti naredbu:php artisan migrate

Ili sledeću u slučaju da bazu izgradite ponovo iz nule.php artisan migrate:refresh

Recimo da smo zaboravili dodati kolonu opis. Šta radimo:

* Izvršimo naredbu:

php artisan make:migration add\_description\_to\_courses\_table --table=courses

* Dodamo u up() i down() funkcije respektivno:

Schema::table('courses', function (Blueprint $table) {

    $table->text('opis')->nullable()->after('ECTS');

});

Schema::table('courses', function (Blueprint $table) {

    $table->dropColumn('opis');

});

### Facker

https://laravel.com/docs/10.x/eloquent-factories

Izvršimo naredbu: php artisan make:factory PostFactory

Dodamo u funkciju definition():

return [

    'naziv' => $this->faker->sentence(3),

    'status' => $this->faker->randomElement(['aktivan', 'neaktivan']),

    'ECTS' => $this->faker->numberBetween(5, 10),

    'opis' => $this->faker->paragraph(3),

];

U datoteku DatabaseSeeder.php u funkciju run() dodamo:

\App\Models\Course::factory()->count(10)->create();

I na kraju pokrenemo:

php artisan db:seed

### Kontroler

php artisan make:controller CourseController

Dodamo metode:

public function index()

{

    $predmeti = Course::all();

    return view('predmeti', ['naziv'=>'Predmeti', 'predmeti'=>$predmeti]);

}

public function show(string $id)

{

    $predmet = Course::where('id', $id)->get();

    dd($predmet);

    return $id;

}

Kreiramo predmeti.blade.php

@extends('master')

@section('title', $naziv)

@section('main-content')

    <h2>Predmeti</h2>

    <ul>

        @foreach($predmeti as $predmet)

        <li><a href="/predmeti/{{$predmet['id']}}">{{$predmet['naziv']}}</a></li>

        @endforeach

    </ul>

@endsection

I dodamo u route:

* Route:: resource('/predmeti', CourseController::class);

Ili

* Route::get('/predmeti', [CourseController::class, 'index']);

Tipovi podataka u tabeli:

https://laravel.com/docs/4.2/schema

Neke važne komande:

php artisan route:list

php artisan make:controller CourseController –resource

php artisan db:seed

php artisan migrate:refresh

php artisan make:model Course –mfc