Patryk Jugowiec

Frameworki PHP

Informatyka
Niestacjonarne
semestr 6

BACKEND

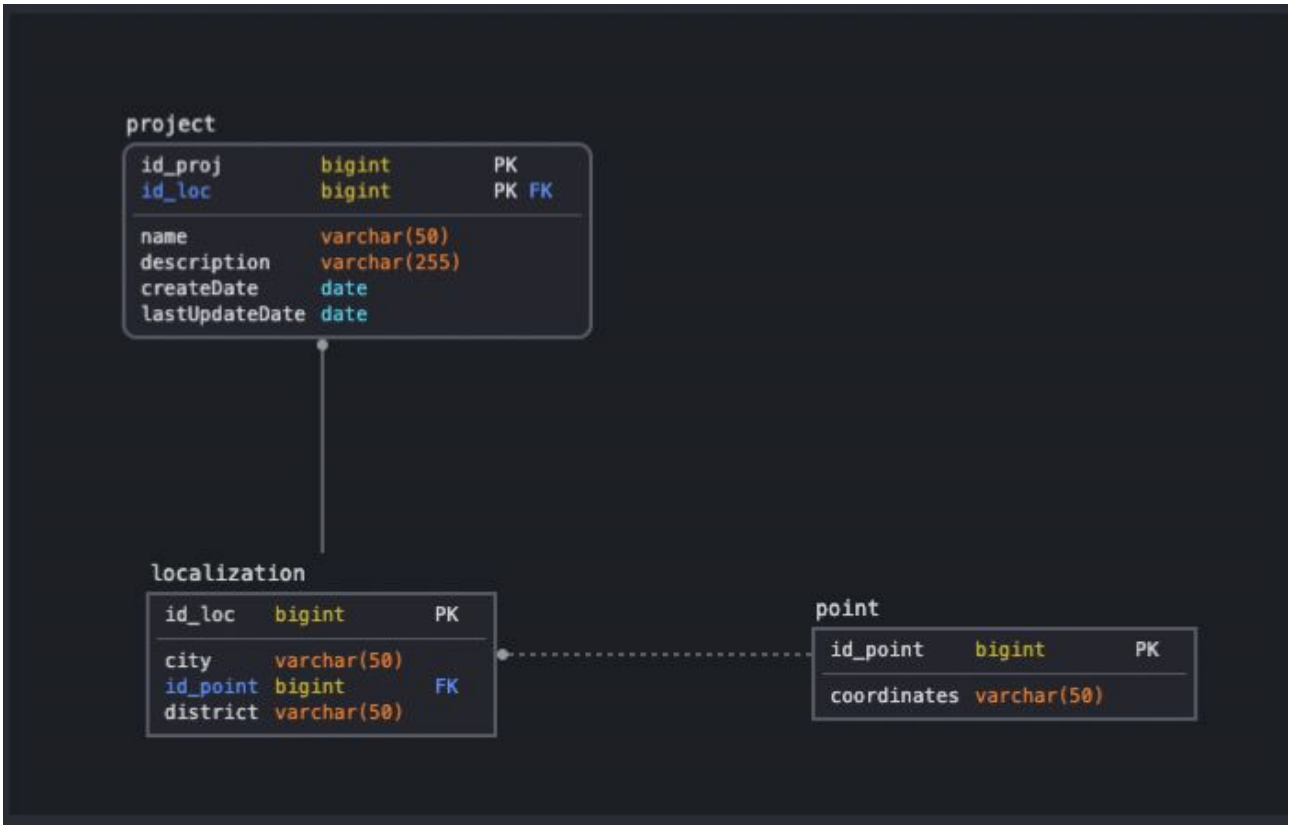
Backend (serwer aplikacji) został zaimplementowany w języku PHP przy pomocy frameworku Symfony w wersji 5. Baza została wygenerowana na podstawie modeli (ORM) w w/w frameworku.

PHP – interpretowany, skryptowy język programowania zaprojektowany do generowania stron internetowych i budowania aplikacji webowych w czasie rzeczywistym.

Symfony – framework dla aplikacji internetowych napisany w języku PHP bazujący na wzorcu projektowym MVC.

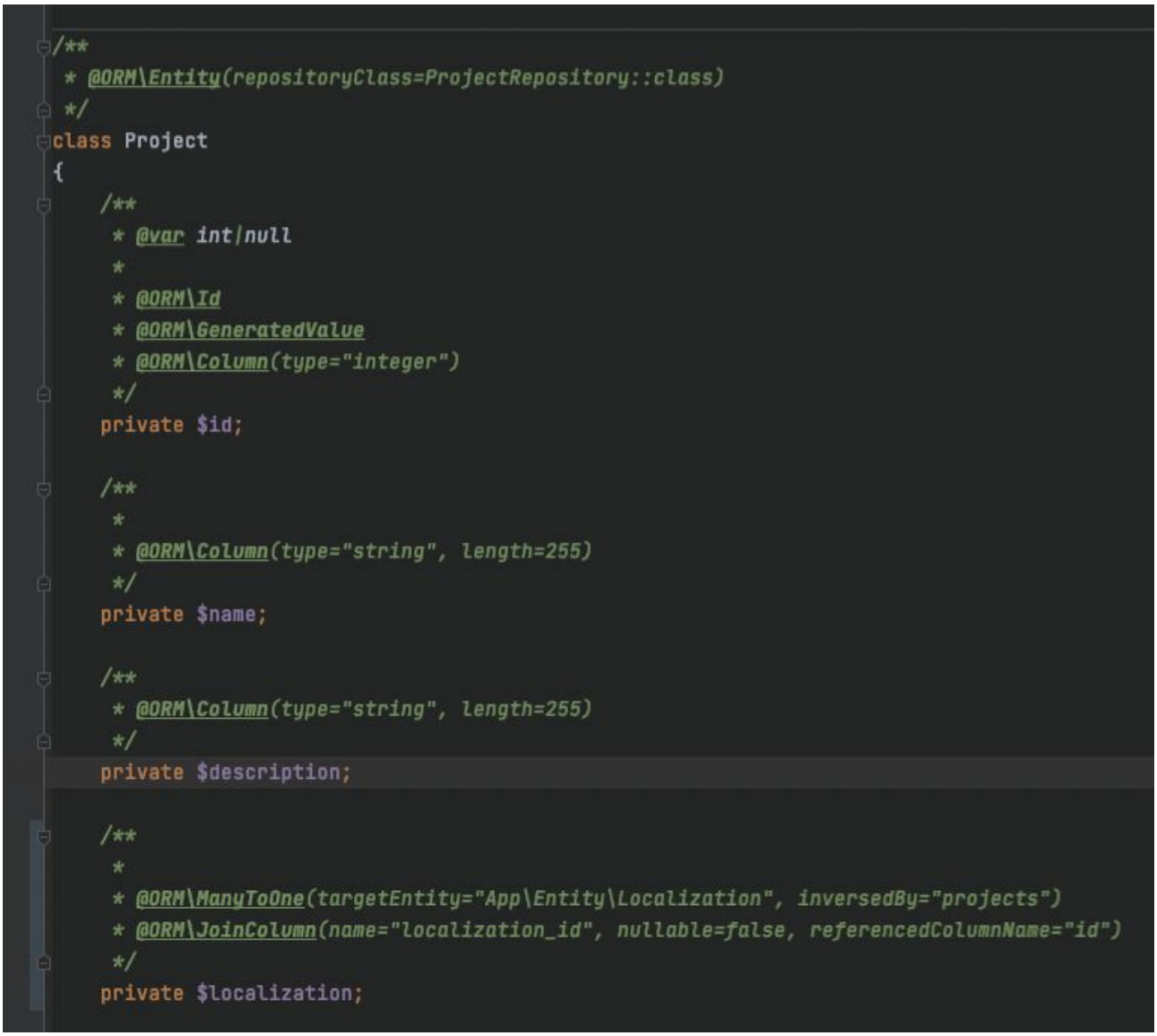
ORM - W informatyce jest techniką programowania służącą do konwersji danych między niekompatybilnymi systemami typów przy użyciu zorientowanych obiektowo języków programowania. W efekcie tworzy się „wirtualna baza danych obiektów”, z której można korzystać z poziomu języka programowania. Dostępne są zarówno darmowe, jak i komercyjne pakiety, które wykonują mapowanie obiektowo-relacyjne, chociaż niektórzy programiści decydują się na tworzenie własnych narzędzi ORM.

Model bazy danych przygotowany w Data Modelerze:



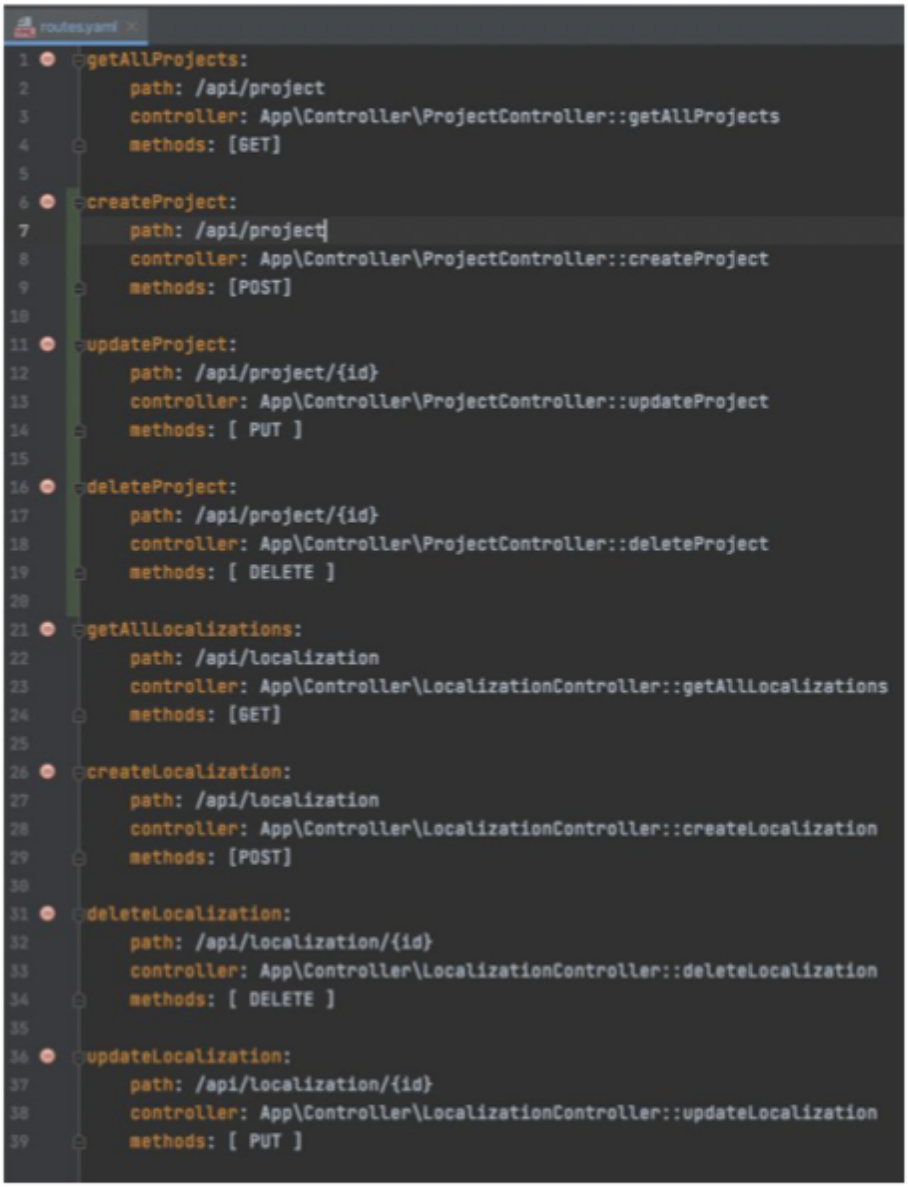
ENCJE:

Encja Projektu na podstawie której framework Symfony wygeneruje odpowiednią tabele. Jak widać na poniższym screenie został wykorzystany **ORM** w którym definiujemy tabele. Widać także połączenie relacyjne do encji “Lokalizacja”. ManyToOne ponieważ jedna lokalizacja może mieć wiele projektów.



```php
/**
 * @ORM\Entity(repositoryClass=ProjectRepository::class)
 */
class Project
{
    /**
     * @var int|null
     *
     * @ORM\Id
     * @ORM\GeneratedValue
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     *
     * @ORM\Column(type="string", length=255)
     */
    private $name;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $description;

    /**
     *
     * @ORM\ManyToOne(targetEntity="App\Entity\Localization", inversedBy="projects")
     * @ORM\JoinColumn(name="localization_id", nullable=false, referencedColumnName="id")
     */
    private $localization;
```

Po stronie serwera zostały konkretne kontrolery który udostępniają połączenie poprzez REST API.



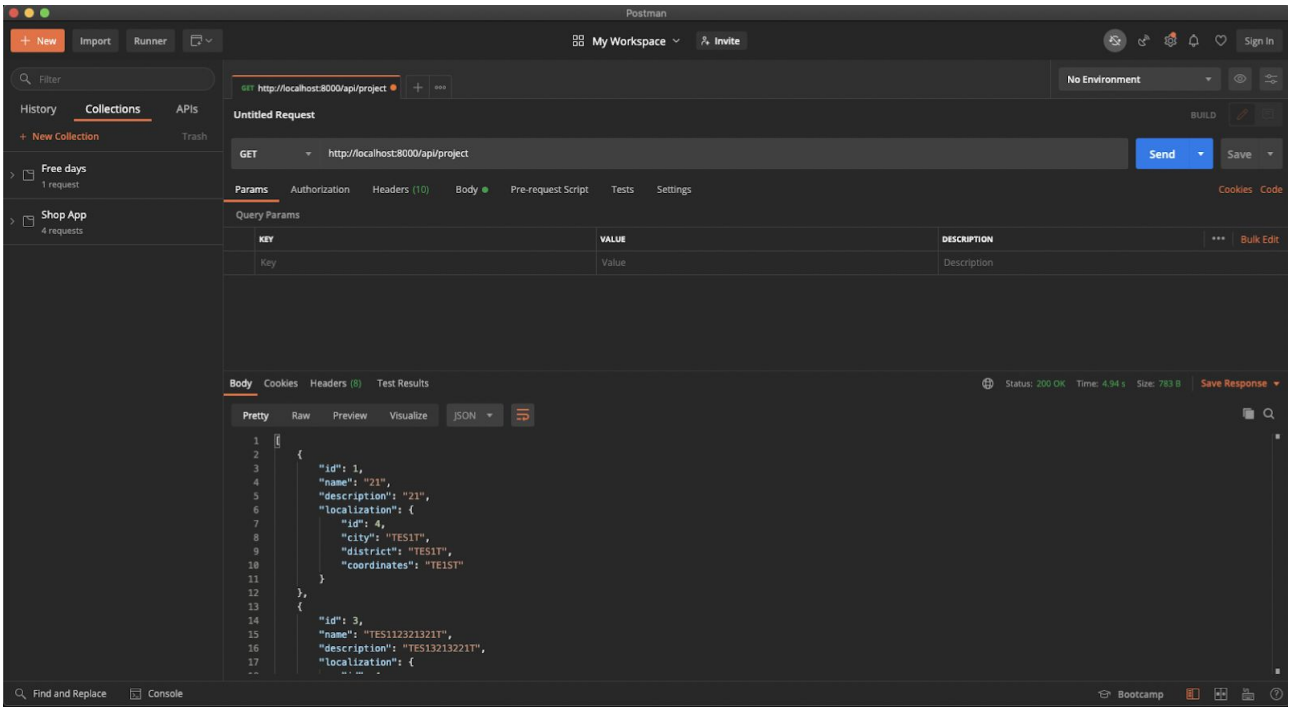
REPOZYTORIUM :

W repozytorium (część logiki odpowiedzialnej za komunikacje z bazą danych), zostały dodane mappery które są odpowiedzialne za dostarczenie do kontrolerów gotowego modelu danych.

```
    public function transform(Project $project): array
    {
        return [
            'id'    => (int) $project->getId(),
            'name' => (string) $project->getName(),
            'description' => (string) $project->getDescription(),
            'localization' => $this->transformLocalization($project->getLocalization())

        ];
    }

    public function transformAll(): array
    {
        $projects = $this->findAll();
        $returnArray = [];

        foreach ($projects as $project) {
            $returnArray[] = $this->transform($project);
        }

        return $returnArray;
    }

    public function transformLocalization(Localization $localization): array
    {
        return [
            'id'    => (int) $localization->getId(),
            'city' => (string) $localization->getCity(),
            'district' => (string) $localization->getDistrict(),
            'coordinates' => (string) $localization->getCoordinates()
        ];
    }
}
```

Przykłady wykorzystania komunikacji REST w praktyce:

/api/project GET



/api/project POST