ELOQUENT ORM – РЕАЛІЗАЦІЯ ШАБЛОНУ ACTIVERECORD У ФРЕЙМВОРКУ LARAVEL

Сафронов Р. О., Абдулов О. Р.
ДДМА, м. Краматорськ

Фреймворк Laravel є одним з багатьох фреймворків, які у своїй основі використовують mody програмування PHP. На сьогоднішній момент часу його популярність швидко зростає, що пов’язано із використанням цілого ряду особливостей, які відрізняють цей фреймворк від інших:

– використання менеджеру залежностей Composer, який дозволяє швидко розгортати проект та контролювати залежності;

– використання dotenv для зберігання важливої та секретної інформації;

– підтримка PSR-4;

– запити форм, їх нескладна обробка та робота з ними та багато інших.

Важливим також є те, що цей фреймворк підтримує найбільш потужну реалізацію шаблону ActiveRecord в PHP Eloquent ORM. Крім звичайних CRUD-операцій в ній є м'яке видалення, області запитів, відносини, методи доступу і мутатори, мутатори дат, спостерігачі моделі і багато іншого.

ORM (англ. Object-relational mapping, об'єктно-реляційна проекція) — технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних» [wiki]. Система об'єктно-реляційного відображення (ORM) Eloquent - реалізація шаблону ActiveRecord в Laravel для роботи з базами даних. Кожна таблиця має відповідний клас-модель, який використовується для роботи з цією таблицею. Моделі дозволяють запитувати дані з таблиць, а також вставляти в них нові записи.

Для роботи з Eloquent ORM спочатку створюється модель Eloquent. Моделі зазвичай розташовуються в директорії проекту app або в іншому місці, в якому працює автозавантажувач composer.json. Всі моделі Eloquent успадковують клас Illuminate\Database\Eloquent\Model. Для роботи з базою даних достатньо тільки в моделі вказати назву таблиці з бази даних (рис. 1).

```
<?php
   namespace App;
   use Illuminate\Database\Eloquent\Model;

   class Article extends Model
     {
        protected $table = 'articles';
```

Рисунок 1 – Фрагмент коду моделі Article.php для роботи з таблицею 'articles'

Цього коду достатньо, щоб контролер міг звертатися до моделі та отримувати дані з бази. На рис. 2 наведено фрагмент коду з контролеру ArticlesController, який дозволяє отримати з таблиці 'articles' останні статті з виводом їх по 10 статей на сторінку.

```php
<?php
   namespace App\Http\Controllers
   use Illuminate\Http\Request;
   use App\Article

   class ArticlesController extends Controller
      {
         public function getPosts()
            {
               $objArt = new Article();
               $articles = $objArt->orderBy('id', 'desc')->paginate(10);
               return view('welcome', ['articles' => $articles]);
            }
```

Рисунок 2 – Фрагмент коду контролеру ArticlesController.php для роботи з моделлю Article.php

Додавання записів в таблицю відбувається через контролер ArticlesController.php з використанням методу addArticle(), який показано на рис. 3.

```php
public function addArticle(Request $request) {
    $article = new Article;
    if ($request->method()=='POST') {
        $article->title = $request->title;
        $article->full_text = $request->full_text;
        $article->save();
        return redirect()->route('articles');
    }
    return view('create_post');
}
```

Рисунок 3 – Фрагмент коду контролеру ArticlesController.php для додавання статті в таблицю Article.php

Відносини між таблицями у Eloquent визначаються як методи у класах моделей. На рис.4 представлені методи, які дозоляють встановити зв’язки між таблицями користувачі-статті (один до багатьох) методом user() та статті-коментарі (одна до багатьох) методом comments().

```php
public function comments()
   {
      return $this->hasMany('App\Comment');
   }
public function user()
   {
      return $this->belongsTo('App\User');
   }
```

Рисунок 4 – Фрагмент коду моделі Article.php для зовнішніх зв’язків

Таким чином, Eloquent ORM є потужним інструментом для роботи із базами даних з використанням фреймворку Laravel. Цей інструмент працювати з базою даних на більш високому рівні та звільняє від деталей нижчого рівня - таких як синтаксис запитів і безпеку.