



web разработка

Models  
Laravel Migration

# Съдържание

Модели

Модели и Ларавел

База данни и Ларавел

Миграции

# Задача

- Направете проект на база данни за проекта
- Създайте базата данни в phpMyAdmin
- Създайте два вида потребители - Администратор и Ученик.
- Отпечатайте профила на потребителя на страница Профил.
- Отпечатайте поставените теми за домашно, които Администраторът на сайта е добавил.
- Добавете функционалност за регистрация/логин на потребителя.
- Всички потребители по подразбиране се добавят като ученици. Има един администратор на сайта, чиято роля е посочена директно в базата данни.

# Model

V

C

# Model

- Работи с базата данни и информацията в нея.
- Методите в моделите обработват потока от информация в приложението - в нужния формат и количество я предоставят на контролера и той от своя страна я прехвърля във видимата част на приложението. Или в обратна посока - поемат постъпващата в приложението потребителска информация и я записват в базата данни. Ако се налага преди това я трансформират.

# Model

- Моделите - отговарят, променят свойствата на класа X.
- Контролерите - отговарят за методите/каква функционалност притежава/ на класа X.

# Model - създаване

```
php artisan make:model Homework
```

# Model

**protected \$fillable** = [данините, които потребителя може да записва в БД,  
изброени като елементи на маси]

# Връзка с базата данни в Laravel

# Връзка с базата данни

Конфигурационния файл, отговарящ за базата данни е `config/database.php`

За да свържем приложението с неговата база данни в `.env`

Popълваме съответните стойности за

`DB_DATABASE=/името на базата данни/`

`DB_USERNAME=root/ потребител/`

`DB_PASSWORD=secret/парола, ако има/`

# Laravel Migration

# Laravel Migration

Миграциите са като GitHub за базата данни, позволяващи на екипа от разработчици лесно да променя и споделя структурата на базата данни на приложението.

Миграциите са съпроводени от т. нар. schema builder, който позволява лесно да се построи конструкцията на базата данни.

Ако някога ви се е налагало да добавяте колона към таблица в базата данни , миграциите решават и такива проблеми.

**Facade Schema** поддържа създаване и промяна на всички видове бази данни, които Laravel поддържа.

**Задача.** Направете проект на нужната база данни за приложението, което разработваме

# Laravel Migration

За да създадете миграция използвайте команда

```
php artisan make:migration create_users_table
```

Тя ще се появи в database/migrations directory.

Всеки файл-миграция съдържа в името си момента, в който е създаден, което позволява на Ларавел да определи реда на миграциите.

**Важно!** Не изтривайте файловете-миграции, така нарушавате реда, който Ларавел следи!

# Laravel Migration

Laravel се придържа към следната конвенция при именуване на

Клас **Homeworks**

Контролер **HomeworksController**

Модел **Homework**

Име на таблицата в БД **homeworks**

*При отклоняване от конвенцията за именуване задължително се прилагат, дефинирани в документацията на Laravel фиксове.*

# Laravel Migration

--table и --create опциите, могат да бъдат използване за индикация дали се създава нова таблица или се прави модификация на вече съществуваща таблица.

Тези опции определят и съдържанието на файла-миграция.

Да пробваме -

**php artisan make:migration create\_courses\_table --create=courses** //създаваме таблица

**php artisan make:migration add\_name\_to\_courses\_table --table=courses** //добавяме колона към съществуваща таблица

Да разгледаме съдържанието на файла-миграция

# Laravel Migration

Миграция за таблица `users` има по подразбиране в Ларавел.

В нея можем да добавяме колони, ако преценим, че ни трябват.

Нека добавим колона `user_role`.

Преди да сме мигрирали/ процес при, който миграционния файл създава таблица в базата данни/ можем да добавим колоната, която ни трябва директно във файла.

След миграция на таблицата, за да добавим колона трябва да използваме командата

```
php artisan make:migration add_user_role_to_users_table --table=users
```

# Laravel Migration - да обобщим

Да изпълним следните стъпки за нашето приложение:

1. Създаваме база данни - laravel\_students
2. Добавяме в .env нужната информация
3. Добавяме в CreateUsersTable миграционен файл колона user\_role
4. Модел User също е създаден при инсталацията на Ларавел, по късно в ще работим в него.
5. Изпълняваме команда php artisan migrate, която създава таблицата users в базата данни
6. Заедно с нея в базата данни се създават и други таблици - прегледайте ги. Те не бива да се трият.

# Laravel Migration - да обобщим

След изпълняването на команда ако се появи грешката

```
D:\xampp\htdocs\students>php artisan migrate
Migration table created successfully.

[Illuminate\Database\QueryException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long
que `users_email_unique`(`email`)

[PDOException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long

D:\xampp\htdocs\students>
```

Изпълняваме - **migrate:rollback** - изтриваме таблиците в базата данни

# Laravel Migration

*След Laravel 5.4 трябва да се съобразим със следното*

## **Дължината на индексите и MySQL / MariaDB**

По подразбиране Laravel използва utf8mb4, което позволява и съхраняване на "emojis" в базата данни.

Ако използвате версия на MySQL, по-стара от 5.7.7 или MariaDB по-стара от 10.2.2, ще трябва да конфигурирате дължината на стринга по подразбиране, генериран при миграция, с цел MySQL да създаде индекси за тях.

# Laravel Migration

Единия от вариантите за това е чрез извикване на метода Schema::defaultStringLength в AppServiceProvider:

```
use Illuminate\Support\Facades\Schema;

/**
 * Bootstrap any application services.
 *
 * @return void
 */
public function boot()
{
    Schema::defaultStringLength(191);
}
```

# Laravel Migration

отново изпълняваме команда

```
php artisan migrate
```

```
D:\xampp\htdocs\students>php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
D:\xampp\htdocs\students>
```

Получаваме съобщение за успешна миграция и създаване на таблици.

# Homework Laravel Migration

- Довършете базата данни на проекта в phpMyAdmin
- Добавете PK, FK, индекси на таблиците