Создание таблиц миграции

## [СОЗДАНИЕ (GENERATING MIGRATIONS)](http://laravel.su/docs/5.2/Migrations#generating-migrations)

Для создания миграции используйте make:migration [Artisan command](http://laravel.su/docs/5.2/artisan):

php artisan make:migration create\_users\_table

Миграция будет помещена в папку database/migrations. Каждое имя файла миграции содержит метку времени, которая позволяет Laravel определять порядок применения миграций.

Опции --table и --create служат для задания имени таблицы и указания будет ли миграция создавать новую таблицу. Проще говоря, эти опции служат для предварительного заполнения генерируемого файла миграции:

php artisan make:migration add\_votes\_to\_users\_table --table=users

php artisan make:migration create\_users\_table --create=users

Для определения иного пути для сохранения файла миграции служит опция --path. Путь должен задаваться относительно корня приложения.

## [СТРУКТУРА МИГРАЦИИ (MIGRATION STRUCTURE)](http://laravel.su/docs/5.2/Migrations" \l "migration-structure)

Класс миграции содержит 2 метода: up и down. up нужен для добавления новых таблиц, столбцов или индексов в БД, тогда как down просто отменяет операции, выполненные в методе up.

Внутри обоих методов вы можете использовать конструктор схем для создания и модификации таблиц. Для изучения всех методов, доступных в конструкторе, обратитесь к документации [check out its documentation](http://laravel.su/docs/5.2/Migrations" \l "creating-tables). Давайте взглянем на пример миграции, которая создает таблицу flights:

<?php

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateFlightsTable extends Migration

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

public function up()

{

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

$table->increments('id');

$table->string('name');

$table->string('airline');

$table->timestamps();

});

}

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

public function down()

{

Schema::drop('flights');

}

}

## [ЗАПУСК МИГРАЦИЙ (RUNNING MIGRATIONS)](http://laravel.su/docs/5.2/Migrations" \l "running-migrations)

Для запуска всех еще незапускавшихся миграций используйте команду migrate. Если вы используете [Homestead virtual machine](http://laravel.su/docs/5.2/homestead), то вы должны запустить ее на вашей VM:

php artisan migrate

Если вы получили сообщение об ошибке "class not found" во время запуска, попробуйте выполнить команду composer dump-autoload и потом вновь выполнить миграцию.

#### Принудительные миграции на продакшене (Forcing Migrations To Run In Production)

Некоторые операции миграции могут привести к потере данных. В целях защиты от запуска таких команд у вас будет запрашиваться подтверждение перед их выполнением. Для выполнения команд без подтверждения укажите флаг force:

php artisan migrate --force

### Откат миграций (Rolling Back Migrations)

Для отката последней выполненной миграции используйте команду rollback. Обратите внимание, что откат отменит операции, выполненные последним запущенным файлом миграции, который может подключать другие файлы миграции:

php artisan migrate:rollback

Команда migrate:reset отменит все миграции:

php artisan migrate:reset

#### Откат/запуск миграции одной командой (Rollback / Migrate In Single Command)

Команда migrate:refresh выполняет откат всех миграций, а затем запускает команду migrate. Команда полезна для пересоздания всей БД:

php artisan migrate:refresh

php artisan migrate:refresh --seed