



Virtual Internship Experience

Laravel: Migrate Database

Migration merupakan salah satu fitur yang terdapat pada framework laravel yang memudahkan kita untuk mengelola *database* secara cepat dan dan mudah

Fitur ini sudah muncul ketika kita melakukan instalasi project laravel untuk mengeceknya bisa melalui perintah `php artisan` pada terminal visual code / git bash, maka jika di scroll pada bagian bawah akan terlihat perintah *migrate* seperti pada gambar dibawah ini

```
migrate
migrate:fresh      Drop all tables and re-run all migrations
migrate:install    Create the migration repository
migrate:refresh    Reset and re-run all migrations
migrate:reset      Rollback all database migrations
migrate:rollback   Rollback the last database migration
migrate:status     Show the status of each migration
```

Pada pembahasan kali ini kita tidak akan membahas keseluruhan fitur *migration* pada laravel

Membuat Migration

Untuk membuat *migration* kita dapat menjalankan perintah

```
php artisan make:migration CreatePostsTable
```

Maka file migration akan muncul pada folder

```
database/migrations/2021_12_23_063732_create_posts_table.php
```

Format file pada *migration* merupakan waktu dimana file migration dibuat

Untuk setiap penulisan format *capitalize each word* pada migration maka secara otomatis akan dikonversi menjadi penamaan dengan *underscore* contoh

`CreatePostsTable` maka laravel akan otomatis melakukan konversi penamaannya menjadi seperti ini `create_posts_table`

Atau penulisan *migration* juga langsung dapat dilakukan dengan cara seperti ini

```
php artisan make:migration create_posts_table
```

Pastikan penulisan format *migration* menggunakan bahasa inggris dengan kata jamak (Contoh : *articles, items, categories*) dan diakhiri dengan nama *_table* apabila migration digunakan untuk membuat tabel baru

Adapun *schema* yang digunakan ketika menggunakan perintah diatas adalah *create* yang artinya *migration* tersebut digunakan untuk membuat *table* baru terlihat pada contoh gambar dibawah ini

```
2021_12_23_063732_create_posts_table.php X
database > migrations > 2021_12_23_063732_create_posts_table.php > PHP IntelliSense > CreatePostsTable > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreatePostsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('posts', function (Blueprint $table) {
17             $table->id();
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::dropIfExists('posts');
30     }
31 }
32
```

Membuat model beserta migration

Kita juga dapat langsung membuat model beserta *migration* dengan menggunakan perintah berikut

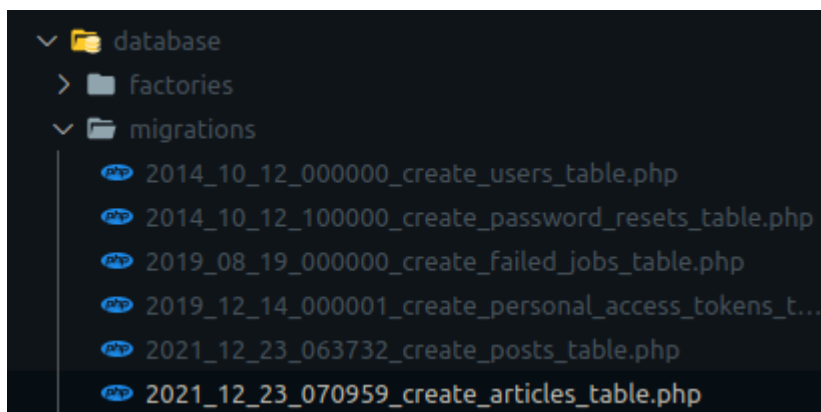
```
php artisan make:model NamaModel -m
```

Sebagai contoh kita akan membuat model Article beserta *migration*-nya

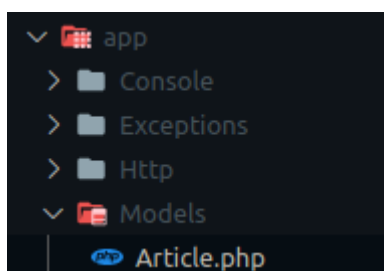
Jalankan perintah berikut pada terminal git bash / visual code

```
temmy@temmy-virtuoso:~/investree-dev$ php artisan make:model Article -m
Model created successfully.
Created Migration: 2021_12_23_152949_create_articles_table
```

Maka akan muncul *migration* seperti pada gambar dibawah ini



Beserta dengan modelnya seperti pada gambar dibawah ini



Menjalankan *Migrate*

Untuk menjalankan *migrate* pastikan kita sudah melakukan *setting* pada file `.env` pada *project* laravel yang sudah ter-*install*

Dapat disesuaikan pada bagian ini

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=root
DB_PASSWORD=root
```

DB_CONNECTION : Digunakan untuk memilih dbms yang akan kita gunakan dapat disesuaikan seperti contoh : mysql, pgsql, sqlite, dll

DB_HOST : *Host* pada *database* yang digunakan di localhost / server

DB_PORT : *Port* yang digunakan pada dbms, pada contoh diatas port 3306 merupakan *port* pada mysql

DB_DATABASE : *Database* yang akan digunakan pada *project* laravel

DB_USERNAME : *Username* yang digunakan pada dbms

DB_PASSWORD : *Password* yang digunakan pada dbms

Kemudian kita bisa sesuaikan *field* yang akan di *set* pada *table* yang akan dijalankan seperti pada contoh gambar dibawah ini

```
2021_12_23_063732_create_posts_table.php X
database > migrations > 2021_12_23_063732_create_posts_table.php > PHP Intelephense > CreatePostsTable > up > #Function#d695c198
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreatePostsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('posts', function (Blueprint $table) {
17             $table->id();
18             $table->string('title');
19             $table->text('content');
20             $table->string('status');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('posts');
33     }
34 }
```

Pada contoh diatas kita melakukan perubahan pada *migration table posts* yang sebelumnya telah kita buat, terdapat penambahan field title dengan tipe data string, content dengan tipe data *text* dan status dengan tipe data *string*

Untuk setiap tipe data sudah otomatis disesuaikan dengan dbms yang digunakan detailnya dapat di cek pada dokumentasi dibawah ini

<https://laravel.com/docs/8.x/migrations#available-column-types>

Kemudian untuk menjalankan *file migration* kita dapat menjalankan perintah berikut pada terminal git bash / visual code

```
php artisan migrate
```

Untuk hasilnya dapat dilihat pada contoh gambar dibawah ini

```
temmy@temmy-virtuoso:~/investree-dev$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (200.72ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (215.75ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (235.54ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (637.18ms)
Migrating: 2021_12_23_063732_create_posts_table
Migrated: 2021_12_23_063732_create_posts_table (147.87ms)
```

Kemudian *table* akan bertambah pada *database* seperti yang tampil pada gambar dibawah ini

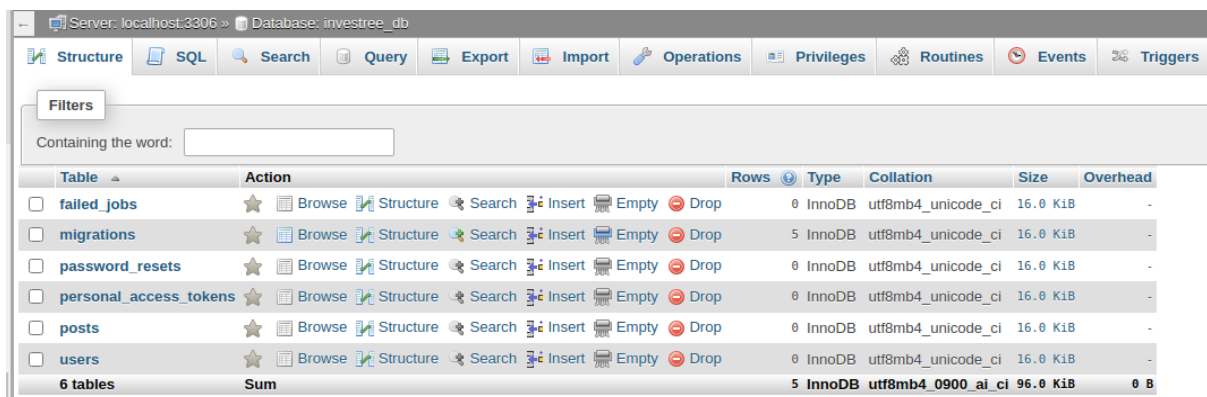
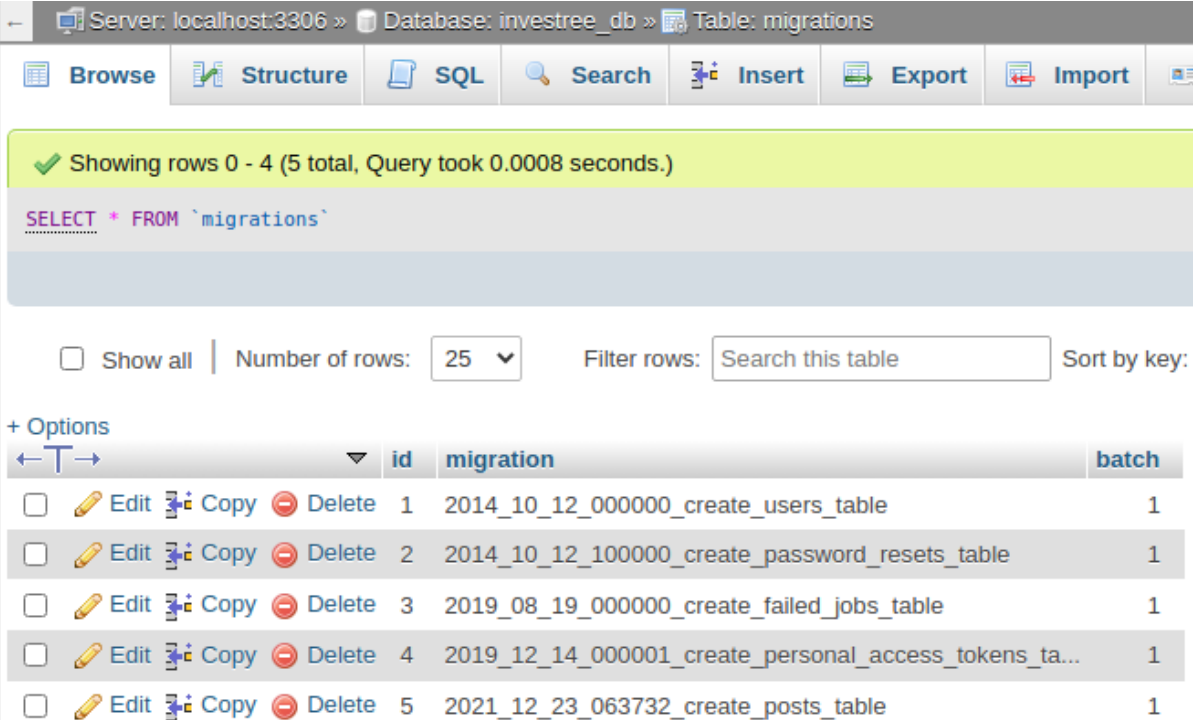


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> posts	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
6 tables	Sum	5	InnoDB	utf8mb4_0900_ai_ci	96.0 KiB	0 B

Table failed_jobs, migrations, password_resets, personal_access_tokens merupakan *table* bawaan yang dimiliki oleh laravel pada saat instalasi pertama kali

Kemudian untuk cara kerja *migration* dijalankan secara *batch* yang menerapkan urutan / atau kondisi ketika *file* tersebut dibuat secara bersamaan dan dijalankan *migrate*, *table migrations* merupakan *table default* yang dimiliki oleh laravel yang berfungsi untuk mencatat semua *log migration* yang sudah dilakukan

Dapat dilihat pada gambar dibawah ini



Server: localhost:3306 » Database: investree_db » Table: migrations

Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

SELECT * FROM `migrations`

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key:

+ Options

				id	migration	batch
<input type="checkbox"/>	Edit	Copy	Delete	1	2014_10_12_000000_create_users_table	1
<input type="checkbox"/>	Edit	Copy	Delete	2	2014_10_12_100000_create_password_resets_table	1
<input type="checkbox"/>	Edit	Copy	Delete	3	2019_08_19_000000_create_failed_jobs_table	1
<input type="checkbox"/>	Edit	Copy	Delete	4	2019_12_14_000001_create_personal_access_tokens_ta...	1
<input type="checkbox"/>	Edit	Copy	Delete	5	2021_12_23_063732_create_posts_table	1

Field migration pada *table migrations* yang melakukan pencatatan nama *migration* yang berhasil dijalankan dan *field batch* yang melakukan pencatatan urutan ketika *file* tersebut dijalankan oleh perintah *migrate*

Menambahkan *field* menggunakan *migration*

Untuk melakukan penambahan *field* pada *table* yang sudah di *migrate*, maka kita perlu menambahkan *migration* baru

```
php artisan make:migration NamaFile --table=nama_table
```

Diatas merupakan cara penulisan apabila kita mau menambahkan *field* baru pada *existing table*

Silahkan jalankan perintah berikut untuk menambahkan *migration field* pada *table posts*

```
php artisan make:migration AddImageFieldToPostsTable --table=posts
```


Dapat dilihat pada contoh gambar dibawah ini

```
temmy@temmy-virtuoso:~/investree-dev$ php artisan make:migration AddImageFieldToPostsTable --table=posts
Created Migration: 2021_12_23_160440_add_image_field_to_posts_table
```

Kemudian kita bisa menambahkan *field image* dengan tipe data *string* seperti pada contoh gambar dibawah ini

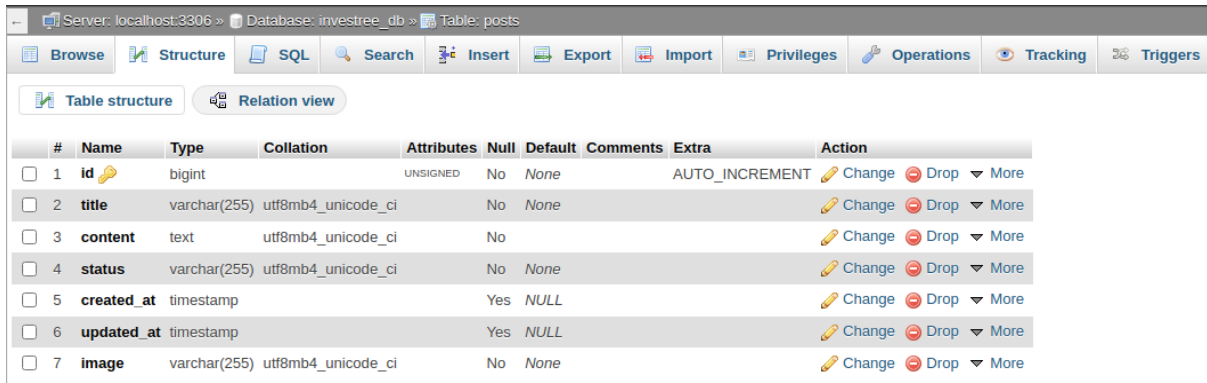
```
2021_12_23_160440_add_image_field_to_posts_table.php ×
database > migrations > 2021_12_23_160440_add_image_field_to_posts_table.php > PHP Intelephense >
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class AddImageFieldToPostsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::table('posts', function (Blueprint $table) {
17             $table->string('image');
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      *
24      * @return void
25      */
26     public function down()
27     {
28         Schema::table('posts', function (Blueprint $table) {
29             $table->dropColumn('image');
30         });
31     }
32 }
```

Untuk melakukan penambahan *field* dapat dilakukan pada *method up()* sesuaikan tipe data field yang dibutuhkan kemudian pada *method down()* digunakan untuk melakukan proses *rollback*

Kemudian jalankan perintah php artisan migrate untuk menambahkan *field* pada *database*

```
temmy@temmy-virtuoso:~/investree-dev$ php artisan migrate
Migrating: 2021_12_23_160440_add_image_field_to_posts_table
Migrated: 2021_12_23_160440_add_image_field_to_posts_table (77.22ms)
```

Dan hasilnya dapat dilihat pada gambar dibawah ini



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	content	text	utf8mb4_unicode_ci		No				Change Drop More
4	status	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	created_at	timestamp			Yes	NULL			Change Drop More
6	updated_at	timestamp			Yes	NULL			Change Drop More
7	image	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More

Terdapat cara untuk mengubah urutan *field* pada *migration* kita dapat menambahkan fungsi `after('nama_field')` namun fungsi ini hanya dapat berjalan pada *database* mysql

```
2021_12_23_160440_add_image_field_to_posts_table.php X
database > migrations > 2021_12_23_160440_add_image_field_to_posts_table.php > PHP Intelephense > AddImageFieldToPostsTable > down
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class AddImageFieldToPostsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::table('posts', function (Blueprint $table) {
17             $table->string('image')->after('content');
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      *
24      * @return void
25      */
26     public function down()
27     {
28         Schema::table('posts', function (Blueprint $table) {
29             $table->dropColumn('image');
30         });
31     }
32 }
33
```

Dapat dilihat pada contoh gambar diatas ini
Dan jalankan *migrate* kembali

Menjalankan Perintah *Migrate Rollback*

Migrate rollback merupakan perintah laravel yang digunakan untuk menghapus *file migration* berdasarkan urutan *batch*

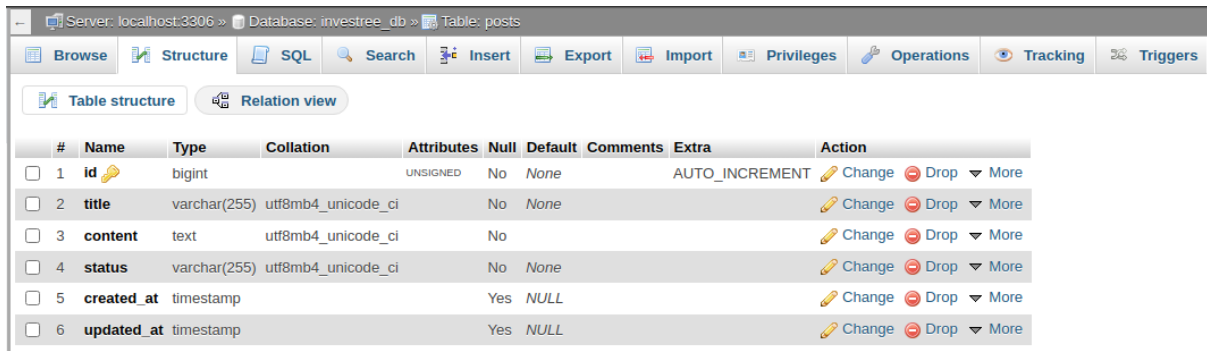
Jalankan perintah berikut untuk menghapus *field image* yang baru saja sudah ditambahkan

```
php artisan migrate:rollback
```

Dapat dilihat pada gambar dibawah ini

```
temmy@temmy-virtuoso:~/investree-dev$ php artisan migrate:rollback
Rolling back: 2021_12_23_160440_add_image_field_to_posts_table
Rolled back: 2021_12_23_160440_add_image_field_to_posts_table (186.64ms)
```

Maka *field image* berhasil terhapus



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	bigint	UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	title	varchar(255)	utf8mb4_unicode_ci	No	None			Change Drop More
<input type="checkbox"/>	3	content	text	utf8mb4_unicode_ci	No				Change Drop More
<input type="checkbox"/>	4	status	varchar(255)	utf8mb4_unicode_ci	No	None			Change Drop More
<input type="checkbox"/>	5	created_at	timestamp		Yes	NULL			Change Drop More
<input type="checkbox"/>	6	updated_at	timestamp		Yes	NULL			Change Drop More