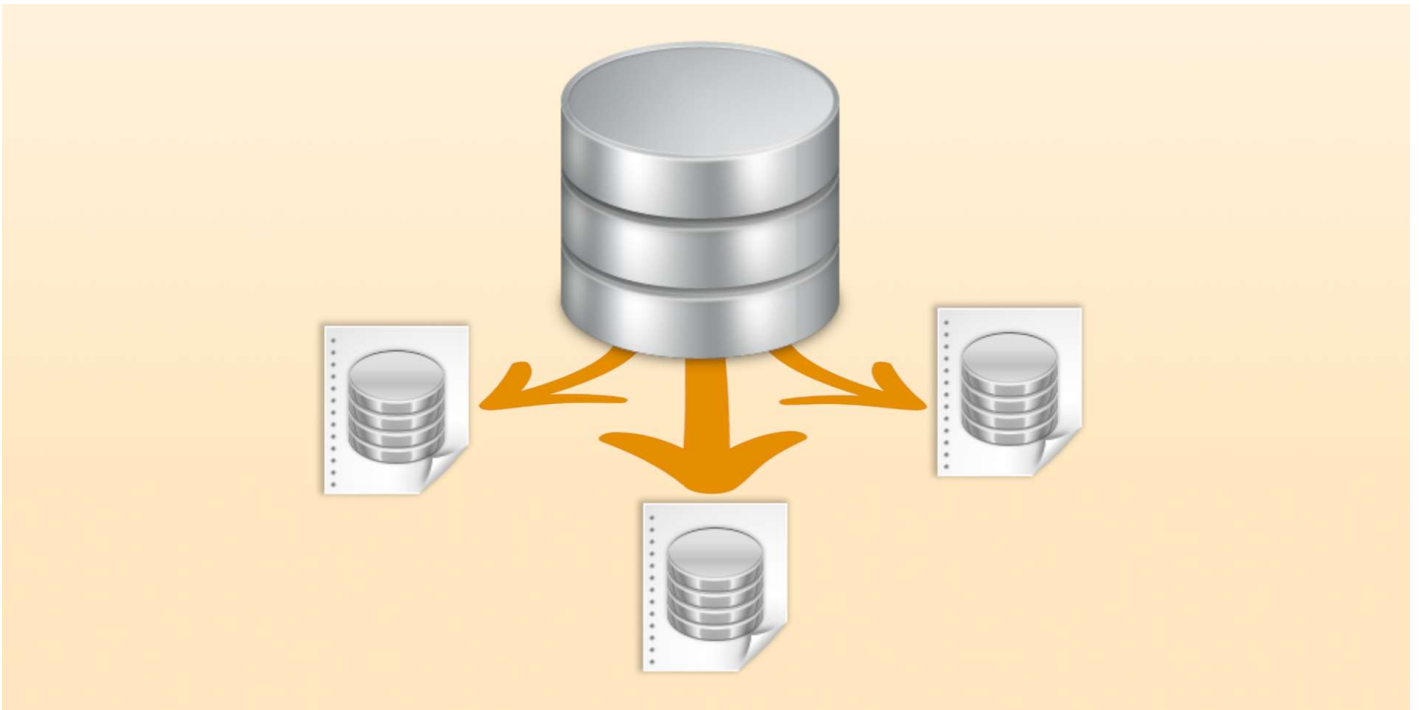


# Laravel : Générer les fichiers de migration d'une BDD existante



## Auteur

[Wilo Ahadi](#)

## Domaine

[Développement d'app. web](#)

## Technologies

[Laravel](#), [PHP](#)

Un guide pour récupérer les structures des tables (migrations) d'une base de données existante dans un projet Laravel 5 et 6

## Sommaire

- [Introduction](#)
- [Installer migrations-generator](#)
- [Utiliser migrations-generator](#)

## Introduction

Dans un projet Laravel, une migration décrit le schéma, la structure d'une table d'une base de données.

On crée un fichier de migration en exécutant la commande *artisan* suivante :

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

Cette commande crée le fichier **2020\_04\_21\_130132\_create\_posts\_table.php** ("2020\_04\_21\_130132" pour moi) dans le répertoire **database/migrations**. Dans ce fichier, nous pouvons décrire la structure de la table «

posts » de la manière suivante :

```
// ...
class CreatePostsTable extends Migration
{

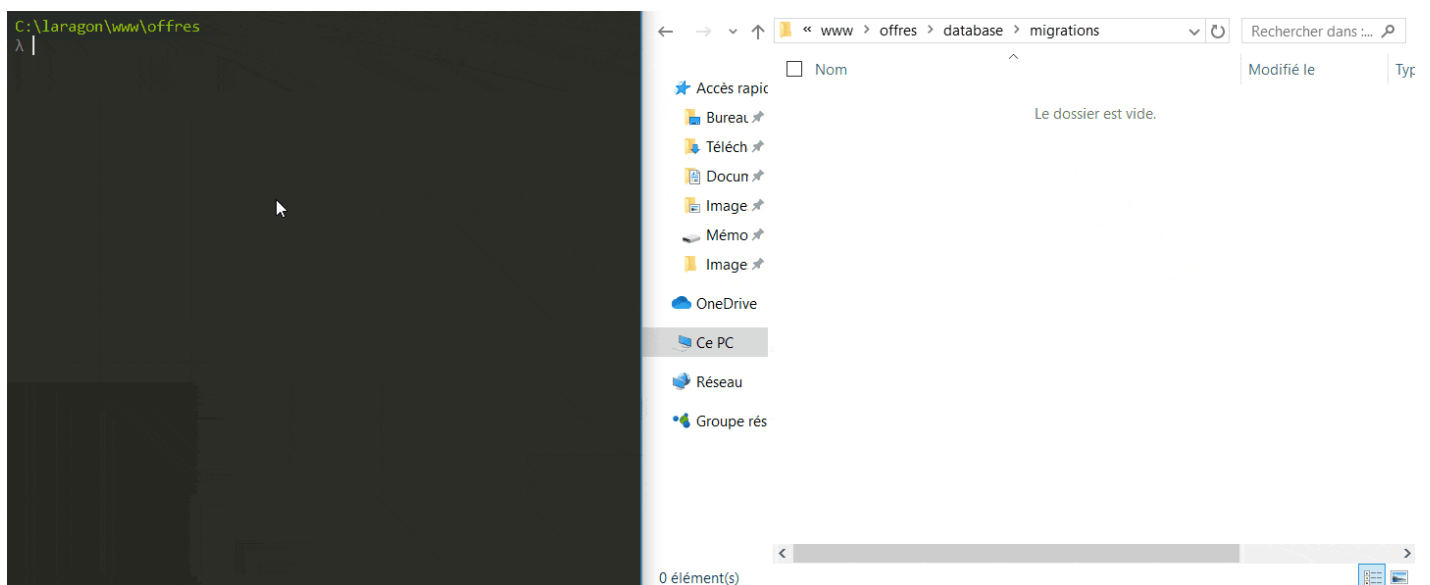
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('title');
            $table->longtext('content');
            $table->timestamps();
        });
    }
    // ...
}
```

Puis pour importer (créer) cette table dans la base de données :

```
php artisan migrate
```

Le processus de migration va dans un seul sens, c'est à dire qu'à partir des fichiers de migration, nous créons (importons) les tables dans la base de données. Mais si nous désirons aller dans l'autre sens : d'une base de données existante, obtenir les fichiers de migration ?

Nous vous proposons le package [Laravel migrations generator](#) qui permet de générer automatiquement les fichiers de migration d'une base de données existante avec toutes ses tables, ses indexes, clés primaires et étrangères.



## Installer migrations-generator

Pour importer migrations-generator avec ses dépendances dans un projet **Laravel 5**, exécutez la commande *composer* suivante :

```
composer require --dev "xethron/migrations-generator"
```

Dans un projet **Laravel 6**, la commande :

```
composer require oscarafdev/migrations-generator 2.0.18
```

Une fois migrations-generator importé, renseignez le Service Provider du package au tableau *\$providers* du fichier **config/app.php** pour les versions de Laravel inférieures à 5.5 :

```
<?php

return [

    // ...,

    'providers' => [
        // ...,
        Way\Generators\GeneratorsServiceProvider::class,
        Xethron\MigrationsGenerator\MigrationsGeneratorServiceProvider::class,
    ],

    // ...

];
```

## Utiliser migrations-generator

Pour créer (générer) les fichiers de migration de toutes les tables de la base de données, exécutez la commande :

```
php artisan migrate:generate
```

Si vous désirez spécifier les tables pour lesquelles générer les fichiers de migration, prenons par exemple les tables « posts », « users » et « comments » uniquement :

```
php artisan migrate:generate posts,users,comments
```

Pour ignorer certaines tables lors de la création des fichiers de migration :

```
php artisan migrate:generate --ignore="users,comments"
```

Et pour l'aide :

```
php artisan help migrate:generate
```

Laravel migrations-generator commence par générer les migrations des tables, leurs colonnes et leurs indexes puis viennent les migrations pour les clés étrangères et les contraintes.