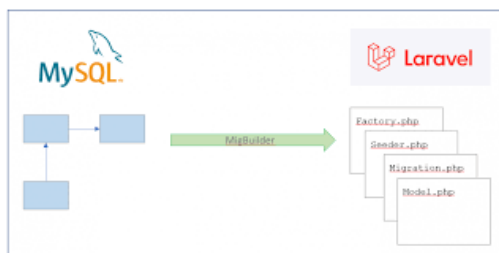


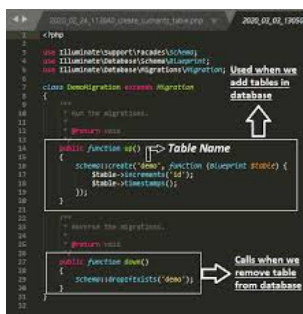
Les Migrations et Seeder dans Laravel

Les migrations et les seeders sont des outils puissants de Laravel qui facilitent la gestion de la structure de la base de données et l'initialisation des données. Voici un aperçu détaillé de leur utilisation :



Migrations

Les migrations sont des fichiers PHP utilisés pour gérer la structure de la base de données. Elles permettent de versionner les changements apportés à la base de données, de sorte que vous pouvez les appliquer (migrer) ou les annuler (rollback) de manière cohérente.



Création d'une migration

Pour créer une migration, vous pouvez utiliser la commande `make:migration` de l'Artisan CLI.

```
php artisan make:migration create_resources_table
```

Cela génère un fichier de migration dans le répertoire `database/migrations`.

Exemple de migration

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateResourcesTable extends Migration
```

```

{
    public function up()
    {
        Schema::create('resources', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->text('description');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('resources');
    }
}

```

Exécution des migrations

Pour appliquer toutes les migrations en attente, utilisez la commande suivante :

```
php artisan migrate
```

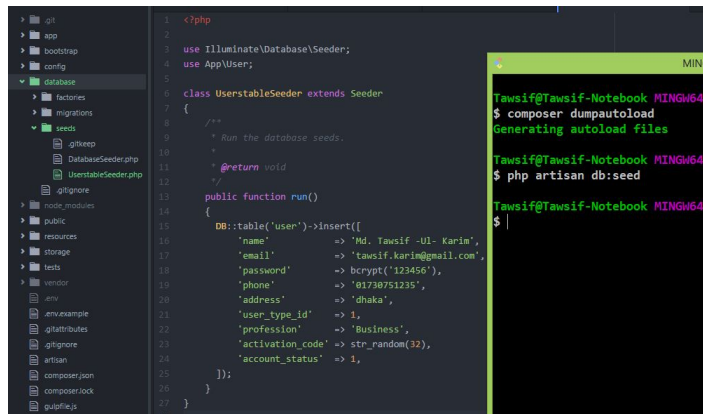
Rollback des migrations

Pour annuler les dernières migrations, vous pouvez utiliser :

```
php artisan migrate:rollback
```

Seeders

Les seeders sont utilisés pour peupler la base de données avec des données initiales ou de test. Ils sont particulièrement utiles pour remplir des tables avec des valeurs par défaut ou des exemples de données.



Création d'un seeder

Pour créer un seeder, utilisez la commande `make:seeder`.

```
php artisan make:seeder ResourceSeeder
```

Cela génère un fichier seeder dans le répertoire `database/seeds`.

Exemple de seeder

```
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;
```

```
class ResourceSeeder extends Seeder
{
    public function run()
    {
        DB::table('resources')->insert([
            'name' => Str::random(10),
            'description' => Str::random(50),
        ]);
    }
}
```

Exécution des seeders

Pour exécuter un seeder, utilisez la commande `db:seed`.

```
php artisan db:seed --class=ResourceSeeder
```

Pour exécuter tous les seeders enregistrés dans le fichier `DatabaseSeeder`, utilisez :

```
php artisan db:seed
```

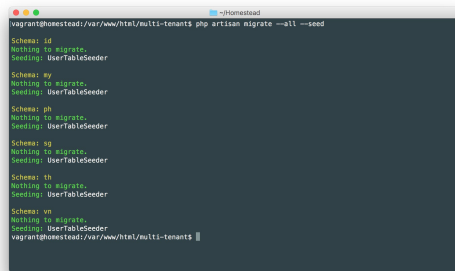
Enregistrement des seeders dans DatabaseSeeder

Le fichier `DatabaseSeeder` se trouve dans le répertoire `database/seeders`. Vous pouvez y enregistrer vos seeders afin de les exécuter tous en une seule commande.

```
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call([
            ResourceSeeder::class,
        ]);
    }
}
```

Exécution des migrations et seeders lors du déploiement



Il est courant d'exécuter les migrations et les seeders lors du déploiement pour s'assurer que la base de données est à jour et contient les données nécessaires.

`php artisan migrate --seed`

Cette commande exécute toutes les migrations et ensuite les seeders enregistrés dans le fichier `DatabaseSeeder`.

Conclusion

Les migrations et les seeders sont des composants essentiels de Laravel qui simplifient la gestion de la structure de la base de données et l'initialisation des données. En utilisant ces outils, vous pouvez facilement versionner, appliquer et annuler les modifications apportées à la base de données, tout en assurant un peuplement cohérent de celle-ci avec des données initiales ou de test.