Laravel

Generateur de Code.

Auteur : ***Berthe Seydou***

07/09/2016

Table des matières

Générateur de Code[: 2](#_Toc459364255)

Exemple pratique [:](#_Toc459364255) 3

Détail d'implémentation[: 2](#_Toc459364255)

Prérequis : Niveau Intermédiaire Laravel.

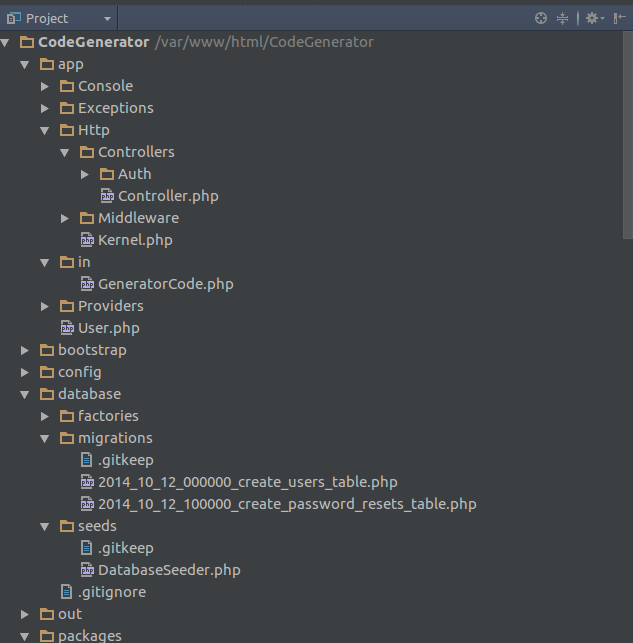
Generateur de Code Laravel:

Dans ce document, il sera question de mettre en place un générateur de code Laravel à partir du **MCD**. En effet on spécifier le MCD sous forme d'un objet et à partir d'une commande (php artisan) on génère les **Controllers**, **Migrations**, **Models,**  **Vues** et même les routes sont aussi générer par rapport au **MCD** et tous ces éléments générer seront placés dans les répertoire appropriés tel que les **migrations** sont placés dans (/database/migrations/), les **Models** dans (/app/) et les **Controllers** dans (/app/Http/Controller).

1. Exemple pratique :

Pennons un exemple pratique pour voir le fonctionnement du générateur avant de le détailler plus (Cet exemple est prise sur une machine Ubuntu 16 mais ça marchera aussi sur Windows).

1- La structure du projet :

Comme on peut le remarquer, il n'y'a pas de Controller, ni Models et Migrations.

2. Specification du MCD :

On s'interesse maintenant au fichier « /app/in/GeneratorCode.php » dans laquelle nous allons specifier notre MCD. Ci-dessous est un example de MCD pour pour ce teste là :

**<?php**

**namespace** App\in;

**use** Berthe\Codegenerator\CallGenerator;

**use** Berthe\Codegenerator\MCD;

*/\*\**

*\* Created by PhpStorm.*

*\* User: seydou*

*\* Date: 01/09/16*

*\* Time: 05:20* م

*\*/*

**class** GeneratorCode **extends** CallGenerator {

**public function** getSite(){

$mcd = **new** MCD();

$mcd->table("film")

->string("titre")

->integer("annee")

->hasMany('acteur')

->end()

->table("acteur")

->string("nom")

->integer("age")

->belongsTo("film")

->end();

**return** $mcd->getSite();

}

}

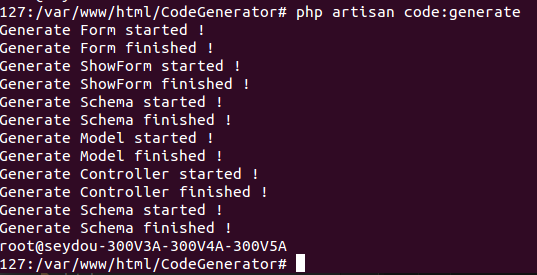
Comme on peut le voir le MCD été spécifier en utilisant un Objet MCD qui est un Constructeur de MCD et possède un ensemble de méthode conçue pour ce but.

Après une bonne spécification du MCD, on génère le Code comme en utilisant la commande : « php artisan code:generate » qui est inclus dans le génerateur de code.

3. Génération du Code :

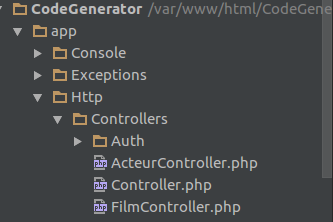
Etant placé à la racine du projet, lancer la commande « **php artisan code:generate** », ceci va générer les différents éléments (Controllers, Migrations, Models et Vues).

Si tous ce passe bien, la commande donne la sortie suivante :



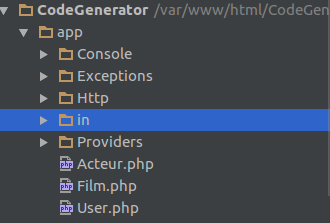
Après l'execution de la commande, voici l'état des différents répertoires pour le répertoire :

→ \App\Http\Controller (Les Controllers) :

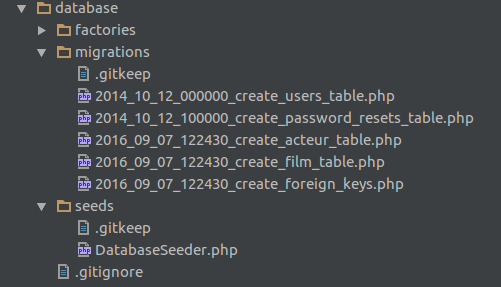


Noter bien les fichiers (**FilmController.php & ActeurController.php**), il représente les Controllers générés.

→ \App (Les Models) :

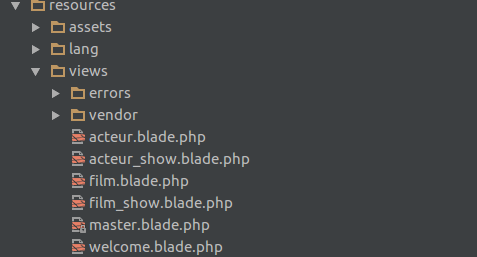


Les fichiers à noter ici sont (**Acteur.php et Film.php**) tout en bas, il représente les Models générés.



Ici on s’intéresse aux trois derniers fichiers du répertoire «**migrations**» qui représente les fichiers de migrations.

→ \resources\views (Les vues).



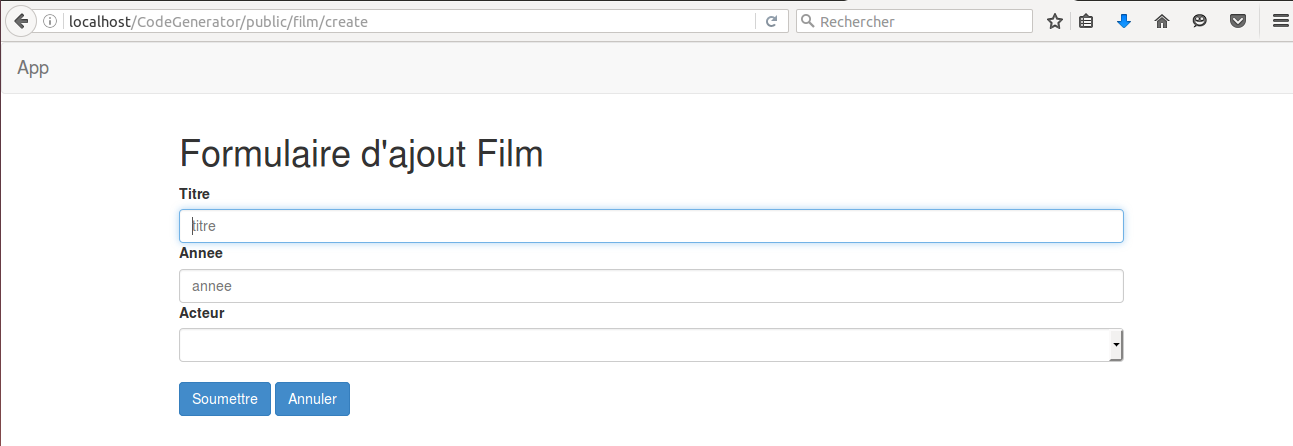
Comme on peut le voir, il y'a 5 fichiers de représentant les vues dans ce cas, deux fichiers pour les formulaires d'ajout, deux fichiers pour l'affichage des tables et aussi et un fichiers « masters » dont tous les autres vues dérives.

Pour tester ces fichiers, il est primordial de faire la migration tous d'abord car il y'a des fichiers de migration générer comme on l'a vu ci-haut, lancer la commande

« **php artisan migrate** » en ligne de commande à la racine du projet. Ceci va créer les différentes table du MCD dans la base de données, assurer vous d'avoir bien configurer la base de données avant de lancer cette commande sinon une erreur sera lancé. (Voir le fichiers **/.env** du projet pour la configuration de la base de données).

Après la génération des migrations on lance teste sur L’URL suivant : **<domaine>/<Projet>/[public]/[film| acteur]/create (Pour voir le formulaire).**

La sortie dans le cas du formulaire d'ajout de Film est la suivante pour les MCD spécifier ci-haut :

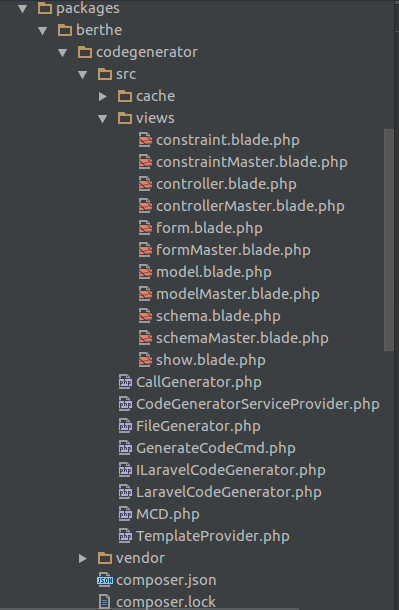


Pour plus détaille référer vous aux fichiers générer.

Passons maintenant au détail d’implémentation.

2. Détail d’implémentation :

Le générateur code est implémenter comme un package **Laravel** pour avoir une grande reutilisabilité. C'est ainsi que nous allons nous intéresser au répertoire « /packages » du projet. Ci-dessous répresente la structure de ce répertoire :



Le répertoire « src » est celui qui est celui qui contient les sources, nous allons nous intéresser plus au contenue de ce répertoire.

→ « **views** » : contient les templates utiliser pour générer les différents fichiers. Par exemple les fichiers (**schemaMaster.blade.php et schema.blade.php**) sont tels que **shcemaMaster.blade.php** contient une squelette d'un fichier de Migration avec les partie redéfinissable et **schema.blade.php** herite de **schemaMaster.blade.php** et rédefinis les differents partie. C'est le même principe pour toutes les autre vues.

→ « **FileGenerator.php**» contient une classe permettant de prendre en paramètre un template blade avec des paramètres et génèrer un fichier de sortie de l'exécution.

→ « **cache** » est un répertoire utiliser par «**FileGenerator.php**» pour stocker des fichiers temporairements et passage de code Blade → Blade → Sortie.

→ « **LaravelCodeGenerator.php** » l’implémentation de du générateur, c'est le chef d’orchestre (noyau) du générateur, il fait appelle à la classe «**FileGenerator**» pour faire la génération de différents fichiers.

→ « **CallGenerator.php** » fait aussi appelle au différentes méthodes de « **CodeGenerator.php**» pour faire la génération, l'utilisateur du package doit hériter de lui pour définir le MCD dans le répertoire « /app/in/GeneratorCode.php ».

→ « **GenerateCodeCmd.php** » : contient l’implémentation de la commande utiliser pour générer le code (**php artisan code:generate**).

→ « **CodeGeneratorServiceProvider.php** » est l’intermédiaire entre le package et l'application **Laravel** utilisant le package, C'est dans ce fichier que tous les classe, et répertoire de l'application sont enregistrer.

Conclusion :

Depot Git du projet : [CodeGenerator](https://github.com/ArianeExtern/generation).

Voir README dans le dépôt pour plus de détail.