

MVC et Laravel - Présentation

L'architecture de Laravel est un exemple typique de l'implémentation du modèle MVC (Model-View-Controller). Voici une caractérisation de l'architecture de Laravel selon ce modèle :

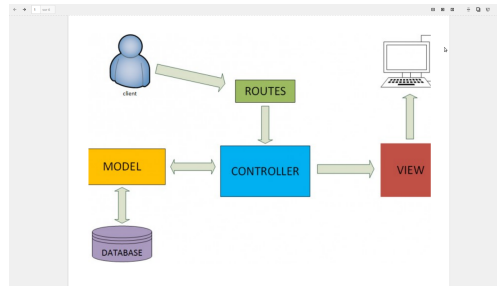
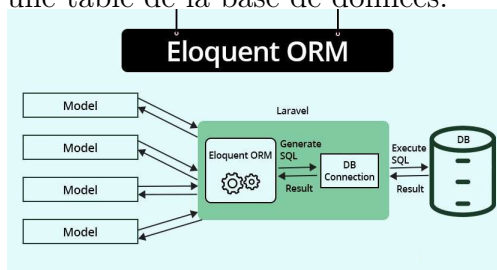


Figure 1: Le Modèle MVC

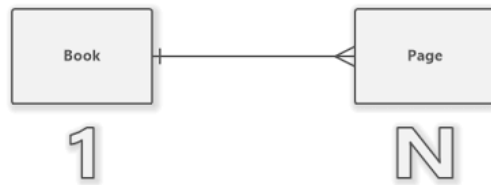
1. Model (Modèle)

Les modèles représentent les données de l'application ainsi que la logique métier. Dans Laravel, les modèles sont généralement des classes Eloquent ORM qui permettent d'interagir avec la base de données de manière simple et expressive.

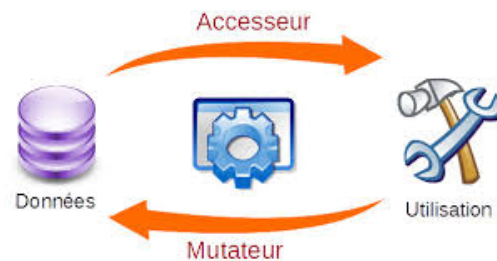
- **Classes Eloquent** : Elles permettent de mapper les tables de la base de données à des objets PHP. Chaque classe de modèle correspond à une table de la base de données.



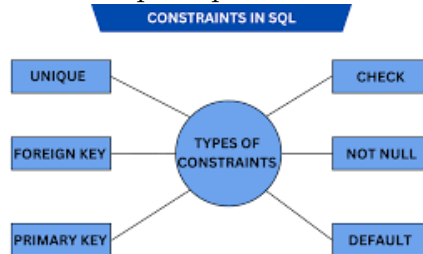
- **Relations** : Laravel facilite la définition de relations entre les modèles (par exemple, un-à-un, un-à-plusieurs, plusieurs-à-plusieurs).



- **Mutateurs et Accessors** : Ils permettent de modifier les attributs des modèles avant de les enregistrer ou de les lire.



- **Scopes** : Ils offrent une manière pratique de définir des conditions de



requête réutilisables.

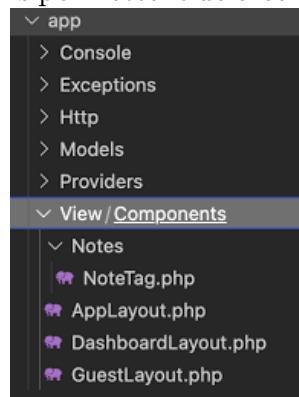
2. View (Vue)

Les vues sont responsables de la présentation des données. Elles reçoivent les données du contrôleur et les affichent à l'utilisateur.

- **Blade Templates** : Laravel utilise Blade, un moteur de templates léger mais puissant. Blade permet d'utiliser des structures de contrôle comme les boucles et les conditions directement dans les fichiers de vue.

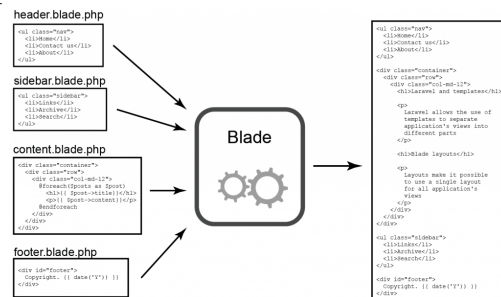


- **Composants de Vue** : Ils permettent de créer des parties réutilisables



de l'interface utilisateur.

- **Layouts** : Ils aident à structurer l'application en utilisant des modèles



de mise en page (layouts) réutilisables.

3. Controller (Contrôleur)

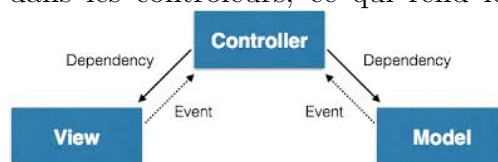
Les contrôleurs traitent les requêtes entrantes, interagissent avec le modèle pour récupérer les données et passent ces données aux vues.

- **Resource Controllers** : Ils offrent une manière standardisée de gérer

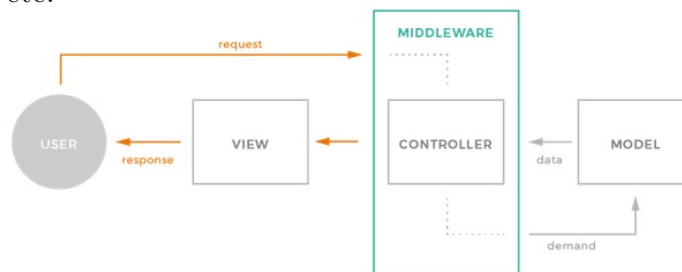
les actions CRUD (Create, Read, Update, Delete).

List #	Title	Author	Language	Price	Sale Price	Actions
2	It	Stephen King	EN	£ 12.99	£ 10.99	Show Edit Delete
3	4.50 from Paddington	Agatha Christie	EN	£ 8.99	£ 6.99	Show Edit Delete
4	Daring Greatly	Brene Brown	EN	£ 16.99	£ 14.99	Show Edit Delete
5	Code	Charles Petzold	EN	£ 11.99	£ ---	Show Edit Delete
6	The Miracle Morning	Hal Elrod	EN	£ 9.99	£ ---	Show Edit Delete
11	The Dome	Stephen King	EN	£ 13.99	£ 11.99	Show Edit Delete
12	Come	Stephen King	EN	£ 6.99	£ ---	Show Edit Delete

- **Dependency Injection** : Laravel facilite l'injection des dépendances dans les contrôleurs, ce qui rend le code plus modulaire et testable.



- **Middleware** : Les contrôleurs peuvent être protégés par des middlewares pour gérer des aspects comme l'authentification, l'autorisation, etc.



Caractéristiques spécifiques de Laravel dans le modèle MVC :

- **Routing** : Laravel utilise un système de routing intuitif qui permet de définir les routes de manière expressive.

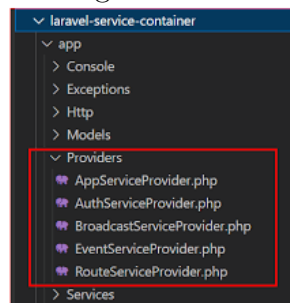
```
fahmida@fahmida-VirtualBox: /var/www/html/laravelpro$ php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	api
					auth:api
	GET HEAD	users	users.index	App\Http\Controllers\UserController@index	web
	POST	users	users.store	App\Http\Controllers\UserController@store	web
	GET HEAD	users/create	users.create	App\Http\Controllers\UserController@create	web
	GET HEAD	users/{user}	users.show	App\Http\Controllers\UserController@show	web
	PUT PATCH	users/{user}	users.update	App\Http\Controllers\UserController@update	web
	DELETE	users/{user}	users.destroy	App\Http\Controllers\UserController@destroy	web
	GET HEAD	users/{user}/edit	users.edit	App\Http\Controllers\UserController@edit	web

- **Validation** : Laravel offre des outils puissants pour la validation des données directement dans les contrôleurs ou via des objets de requête.



- **Service Providers** : Ils jouent un rôle crucial dans l'architecture de Laravel, permettant de configurer et d'initialiser les différents composants de l'application.



- **Artisan CLI** : L'outil en ligne de commande Artisan facilite les tâches courantes comme la création de contrôleurs, de modèles, de migrations de base de données, etc.

```

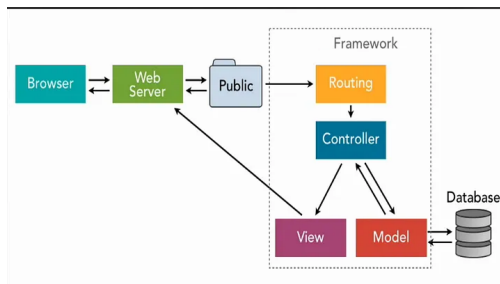
Usage:
  command [options] [arguments]

Options:
  --help (-h)           Display this help message
  --quiet (-q)          Do not output any message
  --verbose (-v|vv|vvv) Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
  --version (-V)        Display this application version
  --ansi                Force ANSI output
  --no-ansi             Disable ANSI output
  --no-interaction (-n) Do not ask any interactive question
  --env                 The environment the command should run under.

Available commands:
  clear-compiled  Remove the compiled class file
  down           Put the application into maintenance mode
  env            Display the current framework environment
  fresh          Remove the scaffolding included with the framework
  help           Displays help for a command
  list           Lists commands
  migrate        Run the database migrations
  optimize       Optimize the framework for better performance
  serve          Serve the application on the PHP development server
  tinker         Interact with your application
  up            Bring the application out of maintenance mode
  app            Set the application namespace

```

Exemple de flux MVC en Laravel :



1. **Requête HTTP** : Un utilisateur envoie une requête HTTP (par exemple, une demande pour afficher une liste d'articles).
2. **Route** : La requête est acheminée vers le contrôleur approprié via le système de routing de Laravel.
3. **Contrôleur** : Le contrôleur traite la requête. Il peut interagir avec un modèle pour récupérer les données nécessaires.
4. **Modèle** : Le modèle interagit avec la base de données pour récupérer ou manipuler les données.
5. **Vue** : Les données sont passées à une vue qui génère la réponse HTML.
6. **Réponse HTTP** : La réponse est envoyée de retour à l'utilisateur.

Conclusion

En résumé, Laravel suit strictement le modèle MVC en fournissant des outils puissants et expressifs pour chaque composant du modèle, facilitant ainsi le

développement de applications web robustes et maintenables.