Le modèle MVC de Laravel

[sommaire]

1. ## Objectifs

- Connaître le contenu des dossiers constituant le Framework Laravel.

- ## Présentation

  - Le **MVC** (**M**odel **V**iew **C**ontroller) est une méthode d'organisation du développement d'applications Web permettant de séparer les différents concepts résultant de nos pages PHP.
  - **Laravel**, comme une grande partie des autres Framework php, a une architecture dite MVC (Model – View – Controller).
  - Le **Model-View-Controller** (MVC) que le framework **Laravel** utilise est essentiellement un modèle architectural qui divise une application en trois composants logiques: **modèle**, **vue** et **contrôleur**, comme son nom l'indique. Ces composants gèrent des aspects de développement spécifiques d'une application pour rendre **Laravel** accessible, puissant et également fournir les outils nécessaires pour exécuter et maintenir des applications volumineuses et robustes.
  - **Laravel** se base effectivement sur le patron de conception MVC, c'est-à-dire modèle-vue-contrôleur.
    - Le **modèle** interagit avec la base de données, les regroupe, traite et gère les données. Le `model` contient les données et leur logique.
    - La **vue** s'occupe principalement de faire afficher ce que le modèle renvoie. Ensuite, elle s'occupe de recevoir toute interaction de l'utilisateur. La `view` contient la présentation graphique à renvoyer à l'utilisateur.
    - Le **contrôleur** prend en charge de synchroniser le **modèle** et la **vue**. Il capte toutes les activités de utilisateur et, en fonction de ces activités, il actionne les changements à effectuer sur l'application. Le `controller` traite les actions utilisateur (via des requêtes), demande au modèle d'effectuer les changements, puis passe les données à la vue.
  - La séparation des composants d'une application en ces trois catégories permet une clarté de l'architecture des dossiers et simplifie grandement la tâche aux développeurs. Ainsi la figure ci dessous nous décris l'architecture MVC de Laravel.

    - **Le model** contient les données et leur logique.
    - **La view** contient la présentation graphique à renvoyer à l'utilisateur.

apcpedagogie.com

- **Le controller** traite les actions utilisateur (via des requêtes), demande au modèle d’effectuer les changements, puis passe les données à la vue.
  Le routing : Bien qu'indépendant de l’architecture MVC, le routing fait partie intégrante de tous les frameworks PHP.

Application MVC avec Laravel

- **Les Modèles (Models)**
  - Ce que nous appelons un Modèle est en réalité un fichier PHP qui ne fait que gérer les échanges avec la base de données. Lorsque nous avons besoin de lire ou écrire dans la base de données, nous faisons appel au Modèle.
  - Le modèle est la couche représentant les données. On l'appellera parfois logique métier.
  - Le modèle consiste en une série de classes. Si les données sont tirées de la BD, chacune des classes représentera une table.
  - Parmi les fonctionnalités codées dans le modèle, on retrouve les relations entre les tables, les accesseurs et modificateurs, les champs calculés, etc.
- **Les Vues (Views)**
  - La vue est constituée de balises HTML qui représentent ce qui sera affiché à l'écran, c'est une interface utilisateur.
  - En plus des balises HTML, la vue peut utiliser des directives et instructions prévues par le moteur d'affichage afin d'effectuer différentes opérations, comme par exemple tester une condition ou encore boucler dans les données fournies par le modèle.
  - La vue pourra faire appel à des ressources externes, comme des feuilles de style, des fichiers JavaScript, des images, etc.
  - Sous Laravel, le moteur d'affichage s'appelle Blade.
- **Les Contrôleurs (Controllers)**
  - Véritable tour de contrôle de notre application, le contrôleur a pour fonction de faire l’interface entre les modèles et les vues. Il est chargé de demander les données par l’intermédiaire des modèles, de traiter ces données et de les transmettre aux vues, prêtes à être utilisées.
- **Le routing**

apcpedagogie.com

- Bien qu'indépendant de l'architecture MVC, le routing fait partie intégrante de tous les Frameworks PHP.
- Dans une architecture classique, nous pointons vers des fichiers :
  - http://monsite.fr/index.php
  - http://monsite.fr/inscription.php
  - http://monsite.fr/login.php
- ...
- Dans une architecture MVC, nous allons pointer vers des dossiers virtuels appelés routes
  - http://monsite.fr/user/inscription
  - http://monsite.fr/user/login
  - http://monsite.fr/blog/article
- ...
- Cette architecture offre de nombreux avantages :
  - Protection des fichiers, ceux-ci n'étant plus affichés par le navigateur
  - Des URLs plus simples à mémoriser pour les utilisateurs
  - Amélioration du référencement si les routes contiennent des mots-clés contenus dans la page correspondante

Exemple de liens entre routes, contrôleurs et vues sous Laravel

- Le schéma ci-dessus montre le cheminement du traitement à partir d'un **URL** jusqu'à l'affichage d'une page Web.
  - On y voit, en rouge, que Laravel commence par rechercher une route qui correspond à l'URL (ici : « / »).
  - On y voit en bleu qu'ensuite, Laravel va appeler le contrôleur indiqué dans le « uses » de la route (ici : PagesController). Plus précisément, en vert, on voit le nom de la méthode d'action à exécuter dans ce contrôler (ici : accueil).
  - En rose, on voit que Laravel va afficher la vue mentionnée dans la méthode d'action. La vue doit être placée dans un sous-dossier de ressources/views (ici : pages) et porter le nom spécifié (ici : accueil).
  - C'est ainsi qu'on obtient à l'écran le visuel de l'URL demandé.

***Source**:https://christianelagace.com/laravel/resume-des-liens-entre-routes-controleurs-et-vues-sous-laravel/*

Application

apcpedagogie.com

- Pour comprendre comment Laravel intègre la logique MVC, créons un exemple de projet Laravel qui affiche la liste des apprenant d'une école de formation.
- Pour créer un nouveau projet, exécutez la commande ci-dessous dans votre terminal:
- `$laravel new` **`gestionApprenant`**

> ***Toutes les commandes artisanales d'un projet Laravel doivent être exécutées à la racine du projet Laravel, vous devez donc revenir à la racine avec la commande `cd` y aller avant d'exécuter la commande. Pour arrêter le serveur, appuyez sur `ctrl + c` sur votre clavier pour Windows***

- **Modèles:**
    - Créer notre modèle de apprenant
    - Créons notre premier modèle, **M** en **MVC**, pour notre application. Comme nous l'avons dit, le modèle s'interface généralement avec un stockage de données comme une base de données **MySQL** par exemple.
    - Dans **Laravel**, le modèle est généralement une classe dont les propriétés correspondent aux colonnes de la base de données.
    - Dans notre base de données, un produit aura les propriétés suivantes:
        - Nom (firstname) - Nom de l'apprenant.
        - Prénom (lastname) - Prénom de l'apprenant.
        - Date Naissance (dteNaissance) - Date Naissance de l'apprenant.
        - Adresse (adress) - Adresse de l'apprenant.
    - Pour créer un modèle dans **Laravel**, exécutez la commande dans votre terminal: **`$php artisan make : model mdlApprenant`**
    - Lorsque vous exécutez cette commande, **Laravel** créera un fichier `mdlApprenant.php` dans le répertoire`app` . Ce sera une classe PHP avec le nom **mdlApprenant** et ce sera le modèle de notre table apprenants dans la base de données.
      https://blog.pusher.com/laravel-mvc-use/

Sommaire du cours Laravel

Source : https://apcpedagogie.com/le-modele-mvc-de-laravel · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com