



Fiche outil

La structure des répertoires d'un projet Laravel

Lors de l'installation de Laravel, ce dernier génère un certain nombre de répertoires. Cette fiche reprends les éléments principaux du cours vidéo associé pour que vous puissiez vous y référer facilement.

```
> app
> bootstrap
> config
> database
> node_modules
> public
> resources
> routes
> storage
> tests
> vendor
```

Table des matières

Le répertoire app.....	3
Le répertoire bootstrap.....	3
Le répertoire config.....	3
Le répertoire database.....	3
Le répertoire public.....	3
Le répertoire resources.....	4
Le répertoire routes.....	4
Le répertoire storage.....	4
Le répertoire tests.....	5
Le répertoire vendor.....	5
Le répertoire app dans le détail.....	6
Le répertoire Console.....	6
Le répertoire Events.....	6
Le répertoire Exceptions.....	7
Le répertoire Http.....	7
Le répertoire Jobs.....	7
Le répertoire Listeners.....	7
Le répertoire Mail.....	7
Le répertoire Notifications.....	7
Le répertoire Policies.....	8
Le répertoire Providers.....	8
Le répertoire Rules.....	8

Le répertoire app

Le répertoire contient tout le code que vous allez générer pour votre applications. Il contient de nombreux autres répertoires. Nous y reviendrons en détail un peu plus loin dans cette fiche outils.

Le répertoire bootstrap

Il contient pour l'essentiel un fichier 'app.php' qui initialise le framework. Il possède également un répertoire 'cache' qui contient des fichiers générés pour l'optimisation des performances comme les fichiers de cache des routes ou des services.

Le répertoire config

Comme son nom l'indique ce répertoire contient l'ensemble des fichiers de configurations de votre application.

Le répertoire database

Il contient toutes les informations concernant vos bases de données. Vous y trouverez les répertoires 'migrations', 'factories' et 'seeds', respectivement responsables :

- du stockage de vos fichiers de migrations (qui vous permettront de créer et modifier la structure de votre base de données),
- du stockage de vos modèles de factories (voir la section de formation dédiée pour en savoir plus à ce sujet)
- du stockage des fichiers permettant d'initialiser du contenu dans vos bases de données

Le répertoire public

Le répertoire 'public' est le point d'entrée unique de votre application par l'intermédiaire du fichier 'index.php'. Cette notion est importante car ce répertoire sera le seul visible de l'extérieur. Cela veut dire que tous les autres répertoires sont sécurisés des attaques de l'extérieur.

Ce répertoire contient également vos différents assets tels que les images, les fichiers CSS et JavaScripts compilés.

Le répertoire resources

Il est important car il contient l'ensemble de vos fichiers de vues avec par défaut le moteur de template de Laravel : [Blade](#). Il contient également vos autres fichiers ressources bruts et non compilés tels que les fichiers LESS, SASS ou JavaScript.

C'est également ici que vous pourrez trouver, ajouter et remplir vos fichiers de langues dans le cas d'une application multilingue.

Le répertoire routes

Il contient plusieurs fichiers listant l'ensemble des routes utilisées par votre application.

Par défaut ces différents fichiers sont :

- **web.php**
 - ✓ accès aux données de session
 - ✓ protection CSRF
 - ✓ cookie protection
- **api.php**
 - ✓ limitation du nombre d'appel (rate limit)
 - ✓ sans état (stateless)
 - ✓ nécessité d'être authentifié par token
 - ✓ pas d'accès aux données de session
- **console.php**
 - ✓ permet la définition des routes pour vos nouvelles commandes dans la console
- **channel.php**
 - ✓ permet la définition des canaux de diffusion d'événements que votre application supporte.

Le répertoire storage

Il va contenir vos fichiers de template Blade compilés, vos fichiers de sessions, les fichiers de cache ainsi que d'autres fichiers générés par le framework.

Il possède plusieurs sous répertoires : app, framework et logs.

Naturellement, le répertoire `app` va contenir les fichiers générés par votre application, tandis que le répertoire `framework` sera responsable du stockage des fichiers générés par le framework lui même, enfin le répertoire `log` contiendra pour sa part les fichiers de logs de votre application. Ces derniers pourront vous être utiles dans la phase de débogage de votre application ou dans le suivi de votre application en production.

Enfin le répertoire `storage/app/public` pourra être utilisé pour stocker les fichiers que génèreront les utilisateurs de votre application, comme les photos de profils par exemple.

Vous devrez pour cela créer un lien symbolique de `public/storage` qui pointerait vers `storage/app/public`.

La commande `php artisan storage:link` vous aidera à réaliser cette opération.

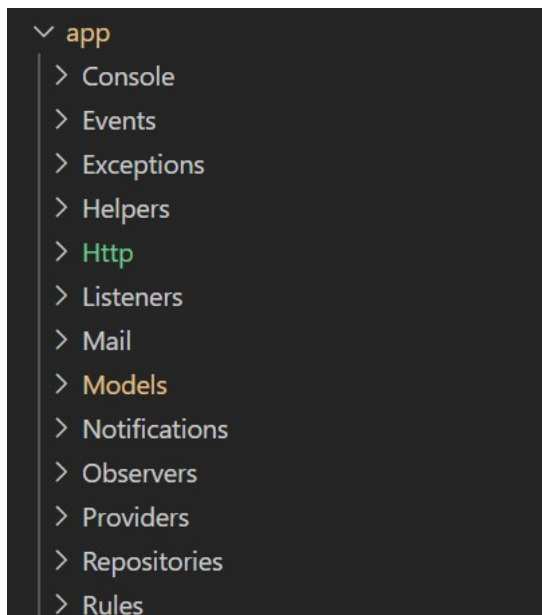
Le répertoire tests

Comme vous l'aurez deviné, ce répertoire contiendra tout vos tests. Laravel est prêt à utiliser d'office PHPUnit.

Le répertoire vendor

Ce répertoire contient les dépendances Composer que vous associerez à votre projet.

Le répertoire app dans le détail



L'essentiel de votre application se trouve dans le répertoire app. Il possède par défaut le namespace App et est auto-chargé par Composer en utilisant le [standard d'autochargement PSR-4](#).

Le répertoire Console

Ce répertoire recevra les commandes Artisan que vous aurez conçues pour votre application en les générant via la commande `make:command`.

Il contient également le fichier `Kernel.php` qui permet l'enregistrement de vos commandes Artisan personnalisées et où sont définies vos [tâches programmées](#).

Le répertoire Events

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `event:generate` et `make:event`.

Il héberge vos classes d'événements (event classes) qui repose sur le pattern [observer](#). Cela permettra à votre application de prévenir d'autres parties de votre application qu'un événement s'est produit.

L'utilisation de ce pattern offre beaucoup de flexibilité et de découplage à votre application.

Le répertoire Exceptions

Il permet la centralisation des différentes exceptions levées par votre application. Le répertoire contient également le gestionnaire de vos exceptions (Handler.php).

Le répertoire Http

Ce répertoire contient tous vos controllers, middlewares et les requêtes de formulaires.

Quasiment toute la logique pour traiter les requêtes entrant dans votre application sera placée dans ce répertoire.

Le répertoire Jobs

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `make:job`.

Il héberge les tâches de votre application pouvant être mises en file d'attente et ainsi permettre à votre application d'améliorer de façon significative ses performances.

Le répertoire Listeners

Attention, tout comme le répertoire Events, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `event:generate` et `make:event`.

Ce répertoire va contenir les classes qui permettront de répondre aux événements (Events) reçus. Ces dernières lanceront des actions en réponses à ces événements.

Le répertoire Mail

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `make:mail`.

Le répertoire Mail va contenir l'ensemble des classes qui gèreront les mails envoyés par votre application.

Le répertoire Notifications

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `make:notification`.

Ce répertoire va contenir l'ensemble de la mécanique permettant de gérer les notifications que votre applications devra envoyer dans son cycle de vie. Les fonctionnalités de notification

de Laravel vous offre une abstraction sur l'envoi de notification et permet d'utiliser indifféremment une multitude de drivers tels que, par exemple, email, Slack, les SMS, ou encore le stockage en base de données.

Le répertoire Policies

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `make:policy`.

Le répertoire des politiques contient les classes de politiques d'autorisation pour votre application. Les stratégies sont utilisées pour déterminer si un utilisateur peut effectuer une action donnée contre une ressource.

Le répertoire Providers

Le répertoire Providers contient tous les fournisseurs de services pour votre application. Les fournisseurs de services amorcent votre application en liant des services dans le conteneur de services, en enregistrant des événements ou en effectuant toute autre tâche pour préparer votre application aux demandes entrantes.

Dans une nouvelle application Laravel, ce répertoire contiendra déjà plusieurs fournisseurs. Vous êtes libre d'ajouter vos propres fournisseurs à cet annuaire selon vos besoins.

Le répertoire Rules

Attention, ce répertoire n'existe pas par défaut. Il est créé uniquement lors de l'utilisation de la commande `make:rule`.

Le répertoire Rules contient les objets de règle de validation personnalisés pour votre application. Les règles sont utilisées pour encapsuler une logique de validation compliquée dans un objet simple.