

Développer en back-end



Développer en back-end

Introduction

A. Découvrir le Framework PHP Laravel

1. Découvrir les notions fondamentales des Frameworks PHP
2. Préparer l'environnement de Laravel

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel
2. Maîtriser le Framework Laravel

C. Approfondir la programmation Laravel

1. Gérer la sécurité
2. Interagir avec la base de données
3. Manipuler l'ORM Eloquent
4. Prendre en charge les tests

D. Administrer un site à l'aide d'un CMS

1. Manipuler les éléments essentiels d'un CMS
2. Personnaliser graphiquement un site à l'aide d'un CMS
3. Manipuler les outils avancés d'un CMS

Conclusion

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

✓ Gestion du routage

- ✓ Utilisation des Middleware (définition, enregistrement, paramétrage, terminate)
- ✓ Protection CSRF
- ✓ Manipulation des contrôleurs
- ✓ Manipulation des requêtes http
- ✓ Manipulation des réponses http
- ✓ Manipulation des vues
- ✓ Création des Template Blade
- ✓ Génération d'URL
- ✓ Manipulation des sessions HTTP
- ✓ Validation des données d'entrée
- ✓ Gestion des erreurs
- ✓ Journalisation (logging)

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage

Qu'est-ce que le routage sous Laravel

- Laravel fournit un système de routes simple. Déclarer une route permet de lier une **URI** (Identifiant de **R**essource **U**niforme, autrement dit la partie de l'adresse qui suit le nom de domaine) à un code à exécuter.
- Le routage est un moyen de créer une URL de requête pour votre application. La meilleure chose à propos du routage Laravel est que vous êtes libre de définir vos routes comme vous le souhaitez.
- Le routage est l'un des composants clés du Framework Laravel, c'est simplement un mécanisme qui effectue le mappage de vos requêtes vers une action de contrôleur spécifique.
- Toutes les routes Laravel sont définies dans le fichier situé sous forme de fichier **/routes/web.php**, qui est automatiquement chargé par le Framework
- Dans Laravel, toutes les demandes sont mappées avec des routes, toutes les routes sont créées dans le dossier racine. Pour votre application Web, vous pouvez définir des itinéraires liés à l'application dans le fichier **web.php** tandis que tous les itinéraires pour l'API sont définis dans le fichier **api.php**.

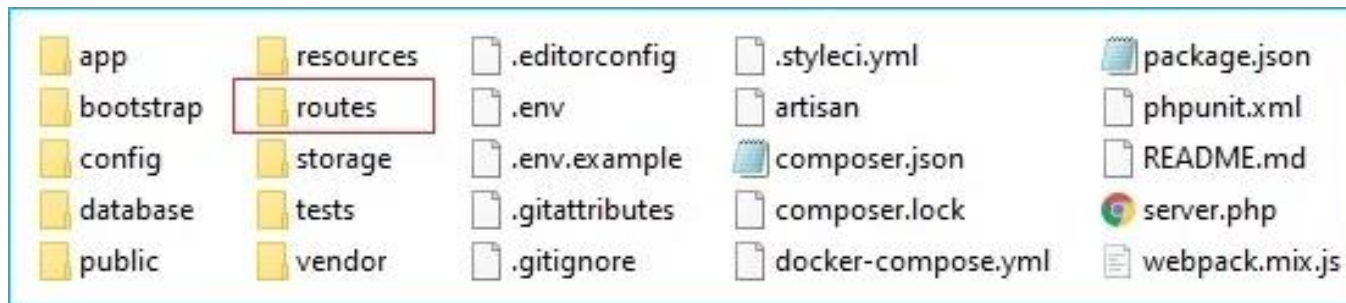
Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage

Présentation



- ✓ Par défaut, Laravel s'installe avec un dossier de routes, pour gérer divers besoins des itinéraires, dans son répertoire racine, ce dossier contient quatre fichiers, **api.php** (utilisé pour gérer les routes de l'API), **channels.php**, **console.php** et **web.php** (gère les routes normales).

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage

Le routage dans Laravel comprend trois catégories:

- ✓ Routage de base
- ✓ Routes nommés
- ✓ Paramètres de Route

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

✓ Routage de base

Le routeur le plus basique de Laravel accepte une URL et retourne une valeur. La valeur peut être **une chaîne**, **une vue** ou **un contrôleur**.

Toutes les routes sont stockées dans **routes/web.php**:

❖ Une Vue

```
<?php

Route::get('/', function () {
    return view('welcome');
});
```

Ce code signifie que lorsque le routeur Laravel reçoit '/', il renvoie une vue appelée «**welcome**» qui est située à **resources/views/welcome.blade.php**.

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

✓ **Routage de base**

❖ **Une Vue**

```
Route::view('/view', 'welcome');
```

Vous pouvez utiliser la **view route** sous laravel si vous voulez retourner uniquement la vue sans rien faire d'autre.

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

❖ Une chaîne

```
<?php

Route::get('/', function () {
    return 'Bonjour tous le monde';
});
```

Lorsque le routeur Laravel reçoit '/', au lieu d'une vue, il renvoie la chaîne « Bonjour tous le monde » .

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

❖ Un contrôleur

La commande pour créer le contrôleur **UserController**:

php artisan make:controller UserController

```
<?php
```

```
Route::get('/user', [UserController::class, 'index']);
```

Ce code signifie que si le routeur reçoit « **/user** », Laravel ira dans la section **UserController** et invoquera la fonction **index**.

Nous parlerons des contrôleurs en détail dans le cour sur la **Manipulation des contrôleurs**

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

✓ Routes nommés

Il est parfois utile de nommer une route, par exemple pour générer une url ou pour effectuer une redirection.

La syntaxe pour nommer une route est celle-ci

```
Route::get('/', function() {           return 'Je suis  
la page d'accueil !';  
})->name('home');
```

Par exemple pour générer l'url qui correspond à cette route on peut utiliser l'helper route :

`Route('home')`

Un avantage à utiliser des routes nommées est qu'on peut réorganiser les urls d'un site sans avoir à modifier beaucoup de code.

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

✓ Paramètres de Route

On peut utiliser un paramètre pour une route qui accepte des éléments variables en utilisant des accolades. Regardez ce code :

```
Route::get('{n}', function($n) {  
    return 'Je suis la page ' . $n . ' !';  
});
```

On dit que la route est paramétrée parce qu'elle possède un paramètre qui peut prendre n'importe quelle valeur.

On peut rendre un paramètre optionnel en lui ajoutant un point d'interrogation mais il ne doit pas être suivi par un paramètre obligatoire. Dans ce cas pour éviter une erreur d'exécution il faut prévoir une valeur par défaut pour le paramètre, par exemple :

```
Route::get('{n?}', function($n = 1) {
```

Le paramètre n est devenu optionnel et par défaut sa valeur est 1.

Développer en back-end

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel

Gestion du routage dans laravel

GET , POST, PUT, PUTCH, DELETE

➤ match

```
Route::match(['get','post'], '/test',function() {  
    return view('welcome');  
});
```

➤ any

```
Route::any('/test',function() {  
    return view('welcome');  
});
```