

Lab 1 :

Installation et organisation, **routage**, **réponses**, ainsi que l'utilisation des **Artisans** et des **contrôleurs** dans Laravel 11. Ce guide est conçu pour être simple, clair et structuré afin de faciliter la compréhension.

1. Installation et Organisation

Avant de commencer à coder, il est important de comprendre comment installer Laravel et organiser votre projet.

Étape 1 : Installation de Laravel

Assurez-vous d'avoir PHP (version ≥ 8.0) et Composer installés sur votre machine. Ensuite, installez Laravel via Composer :

bash

Copier

```
composer create-project laravel/laravel nom-de-votre-projet
```

Naviguez vers le répertoire du projet :

bash

Copier

```
cd nom-de-votre-projet
```

Lancez le serveur de développement intégré :

bash

Copier

```
php artisan serve
```

Accédez à <http://localhost:8000> dans votre navigateur pour vérifier que tout fonctionne correctement.

2. Routage

Les routes permettent de définir les points d'accès de votre application. Par défaut, les routes sont définies dans le fichier `routes/web.php`.

Exemple de Routage Basique

Créez une route simple qui renvoie une réponse texte :

php

Copier

```
// routes/web.php
```

```
use Illuminate\Support\Facades\Route;
```

```
Route::get('/hello', function () {  
    return 'Hello, Laravel!';  
});
```

Accédez à <http://localhost:8000/hello> pour voir la réponse.

ROUTAGE AVEC PARAMÈTRES

Vous pouvez également ajouter des paramètres dynamiques dans vos routes :

php

Copier

```
Route::get('/greet/{name}', function ($name) {  
    return "Hello, $name!";  
});
```

Accédez à <http://localhost:8000/greet/John> pour obtenir "Hello, John!".

3. Réponses

Laravel propose plusieurs façons de retourner des réponses à partir de vos routes ou contrôleurs.

Réponse Texte

php

Copier

```
Route::get('/text', function () {  
    return response('Hello World', 200)  
        ->header('Content-Type', 'text/plain');  
});
```

Réponse JSON

php

Copier

```
Route::get('/json', function () {  
    return response()->json([  
        'message' => 'Hello, JSON!',  
        'status' => 'success'  
    ]);  
});
```

Vue (Blade Template)

Créez un fichier `resources/views/welcome.blade.php` contenant :

html
Copier

```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome</title>
</head>
<body>
  <h1>Bienvenue dans Laravel !</h1>
</body>
</html>
```

Puis, retournez cette vue depuis une route :

php
Copier

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

4. Artisan et les Commandes

Artisan est l'outil en ligne de commande fourni par Laravel pour automatiser diverses tâches.

Création d'une Route avec Artisan

Au lieu de modifier manuellement `routes/web.php`, vous pouvez utiliser Artisan pour générer des routes :

bash
Copier

```
php artisan make:controller WelcomeController --resource
```

Cela crée un contrôleur avec des méthodes standard pour les ressources.

Liste des Routes

Pour afficher toutes les routes disponibles dans votre application :

bash
Copier

```
php artisan route:list
```

5. Contrôleurs

Les contrôleurs permettent de séparer la logique métier des routes. Voici un exemple de création et d'utilisation d'un contrôleur.

Étape 1 : Création du Contrôleur

Générez un nouveau contrôleur avec Artisan :

```
bash  
Copier
```

```
php artisan make:controller UserController  
Cela crée un fichier app/Http/Controllers/UserController.php.
```

Étape 2 : Ajout de Méthodes dans le Contrôleur

Modifiez le fichier `UserController.php` pour ajouter une méthode :

```
php  
Copier
```

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
  
class UserController extends Controller  
{  
    public function show($id)  
    {  
        return "Affichage de l'utilisateur avec l'ID : $id";  
    }  
}
```

Étape 3 : Liaison du Contrôleur à une Route

Dans `routes/web.php`, reliez la route au contrôleur :

```
php  
Copier
```

```
Route::get('/user/{id}', [UserController::class, 'show']);
```

Accédez à <http://localhost:8000/user/123> pour voir "Affichage de l'utilisateur avec l'ID : 123".