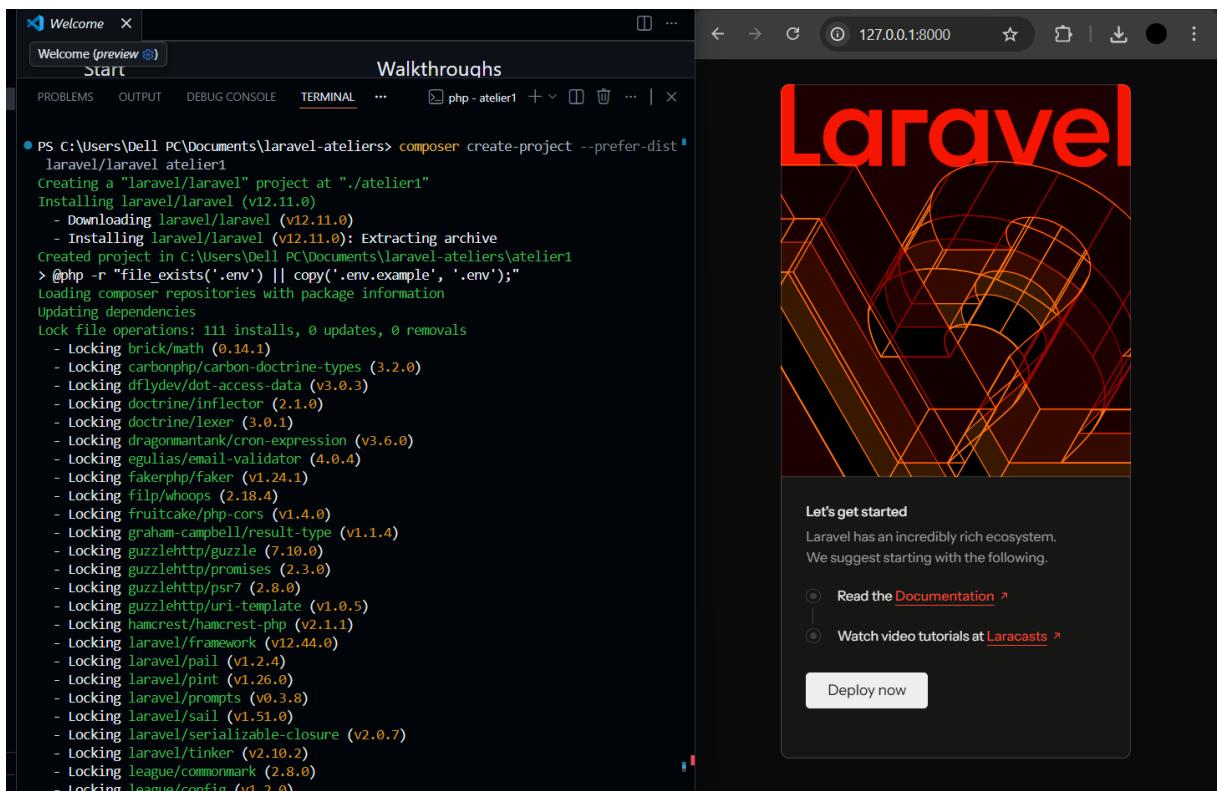


# Atelier 1 Laravel: Introduction à Laravel

## 1. Créer un projet Laravel

2. Exécutez la commande suivante pour créer un projet Laravel  
(remplacez NomDuProjet par le nom que vous souhaitez donner à votre projet) :



The terminal window shows the following command and its execution:

```
PS C:\Users\DELL PC\Documents\laravel-ateliers> composer create-project --prefer-dist laravel/laravel atelier1
Creating a "laravel/laravel" project at "./atelier1"
Installing laravel/laravel (v12.11.0)
  - Downloading laravel/laravel (v12.11.0)
  - Installing laravel/laravel (v12.11.0): Extracting archive
Created project in C:\Users\DELL PC\Documents\laravel-ateliers\atelier1
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
  - Locking brick/math (0.14.1)
  - Locking carbonphp/carbon-doctrine-types (3.2.0)
  - Locking dflydev/doctrine-access-data (v3.0.3)
  - Locking doctrine/inflexor (2.1.0)
  - Locking doctrine/lexer (3.0.1)
  - Locking dragonmantank/cron-expression (v3.6.0)
  - Locking egulias/email-validator (4.0.4)
  - Locking fakerphp/faker (v1.24.1)
  - Locking filp/whoops (2.18.4)
  - Locking fruitcake/php-cors (v1.4.0)
  - Locking graham-campbell/result-type (v1.1.4)
  - Locking guzzlehttp/guzzle (7.10.0)
  - Locking guzzlehttp/promises (2.3.0)
  - Locking guzzlehttp/psr7 (2.8.0)
  - Locking guzzlehttp/uri-template (v1.0.5)
  - Locking hamcrest/hamcrest-php (v2.1.1)
  - Locking laravel/framework (v12.44.0)
  - Locking laravel/pail (v1.2.4)
  - Locking laravel/pint (v1.26.0)
  - Locking laravel/prompts (v0.3.8)
  - Locking laravel/sail (v1.51.0)
  - Locking laravel/serializable-closure (v2.0.7)
  - Locking laravel/tinker (v2.10.2)
  - Locking league/commonmark (2.8.0)
  - Locking league/config (v1.2.0)
```

The browser window shows the Laravel welcome screen with the text "Let's get started" and links to "Documentation" and "Laracasts".

4. Démarrez le serveur de développement Laravel :

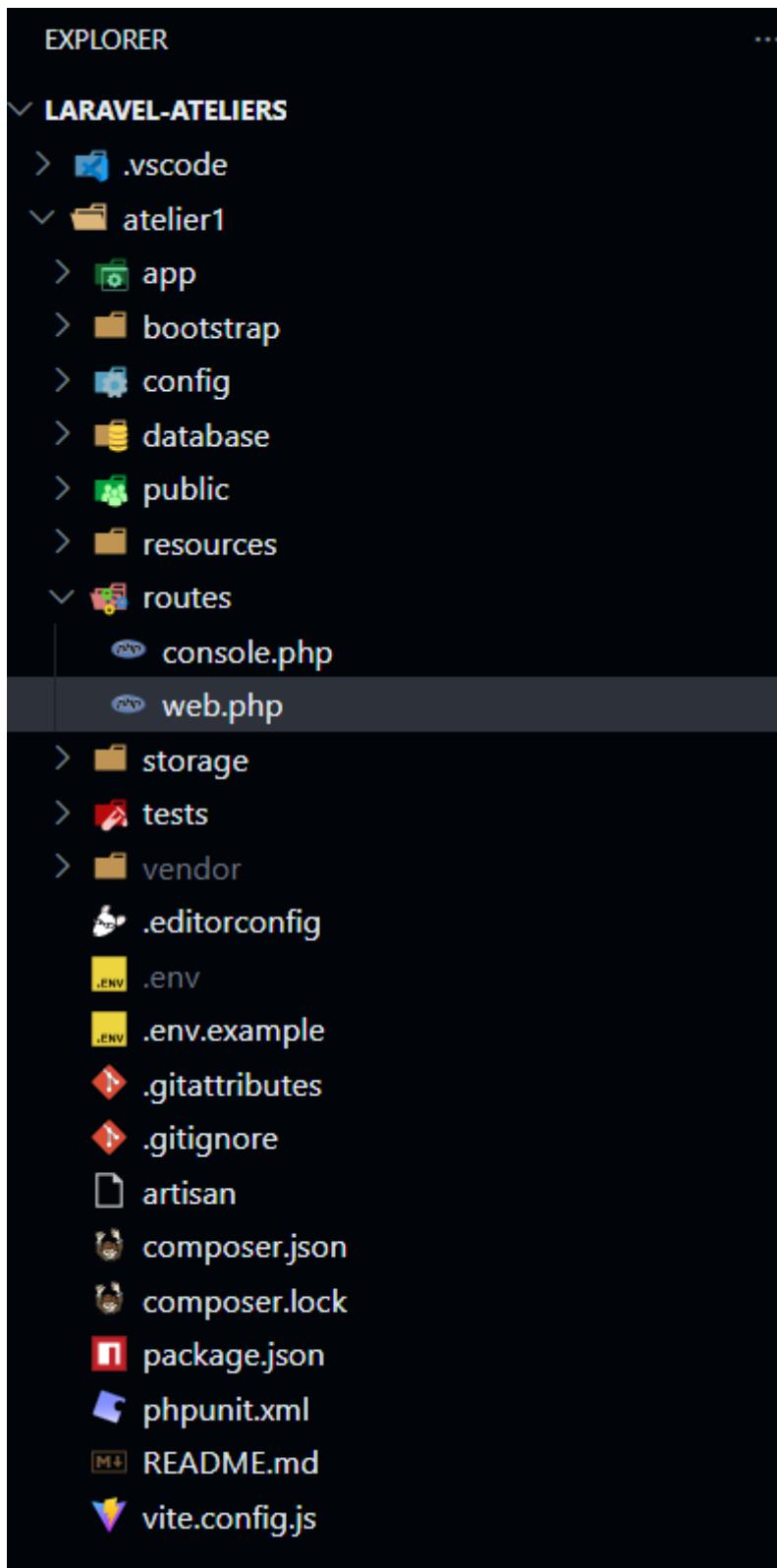


```
PS C:\Users\DELL PC\Documents\laravel-ateliers\atelier1> php artisan serve
```

INFO Server running on [http://127.0.0.1:8000].  
INFO Server running on [http://127.0.0.1:8000].

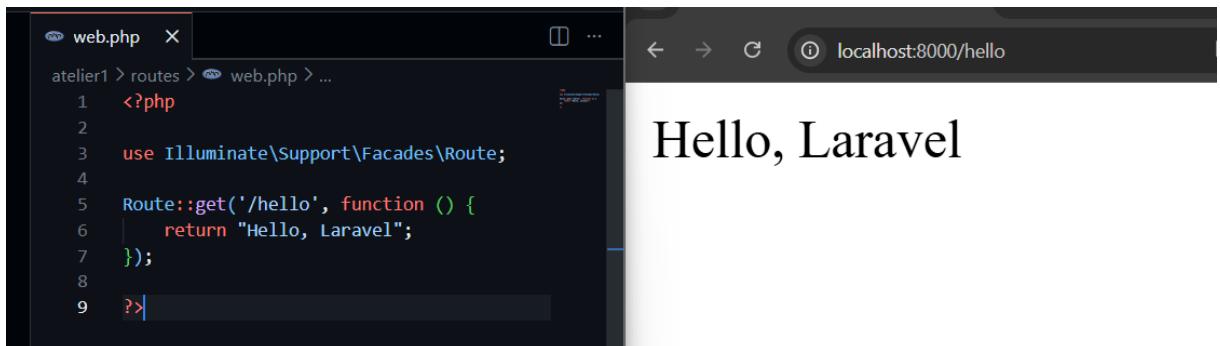
Press Ctrl+C to stop the server

## 2. Architecture du projet Laravel et exploration des dossiers



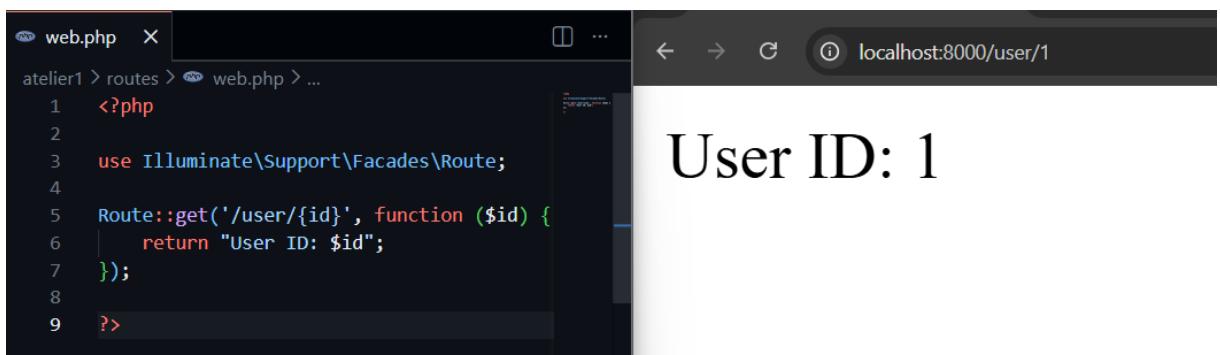
### 3. Créer et tester des routes dans Laravel

#### 3.1 Routes de base



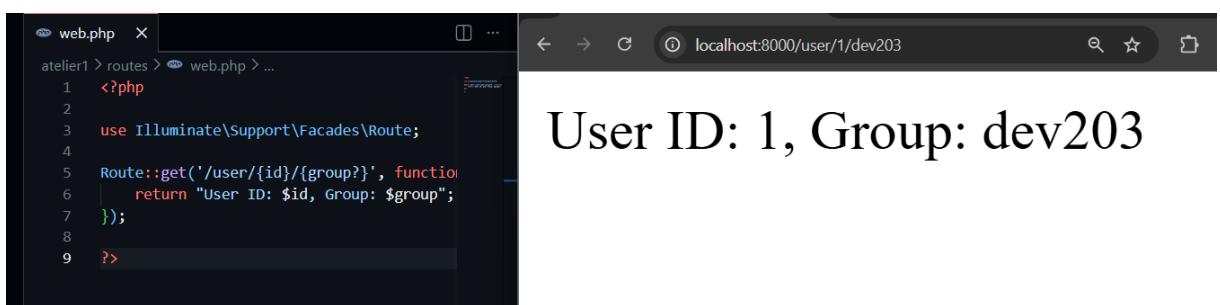
```
<?php
use Illuminate\Support\Facades\Route;
Route::get('/hello', function () {
    return "Hello, Laravel";
});
```

#### 3.2 Route dynamique avec des paramètres



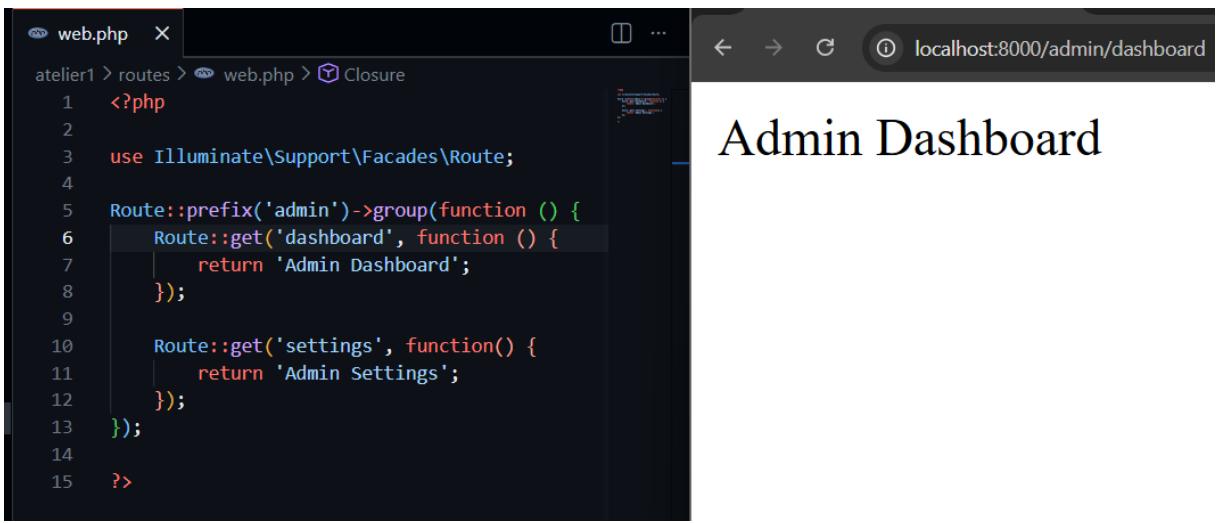
```
<?php
use Illuminate\Support\Facades\Route;
Route::get('/user/{id}', function ($id) {
    return "User ID: $id";
});
```

#### 3.3 Route avec un paramètre optionnel



```
<?php
use Illuminate\Support\Facades\Route;
Route::get('/user/{id}/{group?}', function ($id, $group) {
    return "User ID: $id, Group: $group";
});
```

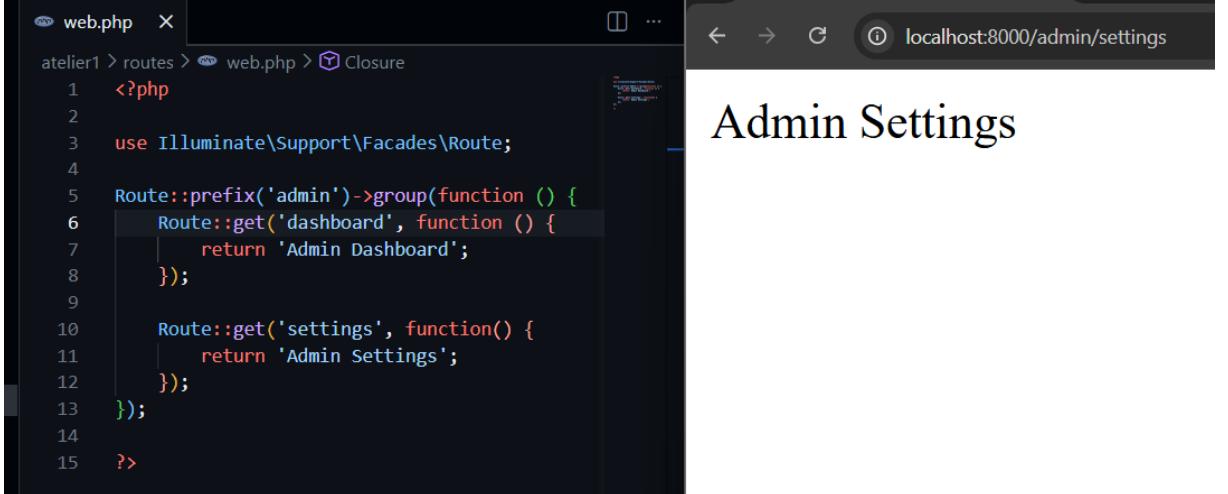
### 3.4 Route préfixée



The screenshot shows a code editor with a file named `web.php`. The code defines a route group under the prefix `'admin'`:

```
<?php  
use Illuminate\Support\Facades\Route;  
  
Route::prefix('admin')->group(function () {  
    Route::get('dashboard', function () {  
        return 'Admin Dashboard';  
    });  
  
    Route::get('settings', function() {  
        return 'Admin Settings';  
    });  
});  
?>
```

The browser address bar shows `localhost:8000/admin/dashboard`, and the page content is "Admin Dashboard".

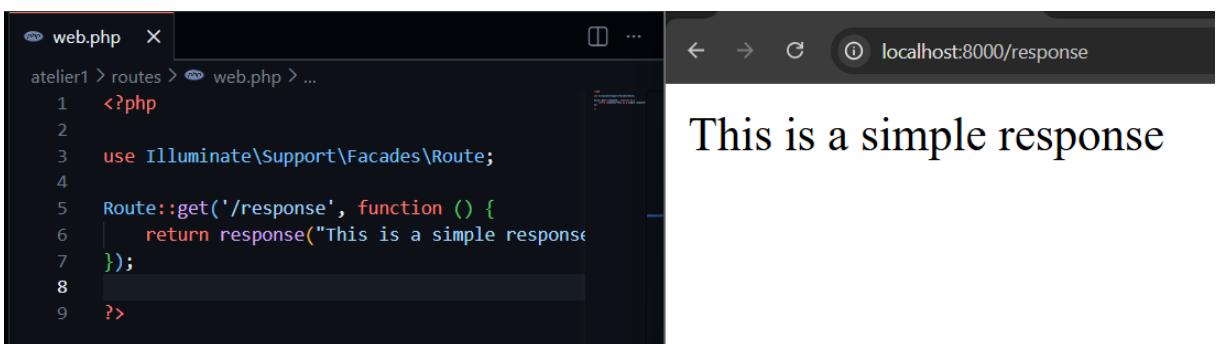
The screenshot shows a code editor with a file named `web.php`. The code defines a route group under the prefix `'admin'`:

```
<?php  
use Illuminate\Support\Facades\Route;  
  
Route::prefix('admin')->group(function () {  
    Route::get('dashboard', function () {  
        return 'Admin Dashboard';  
    });  
  
    Route::get('settings', function() {  
        return 'Admin Settings';  
    });  
});  
?>
```

The browser address bar shows `localhost:8000/admin/settings`, and the page content is "Admin Settings".

## 4. Gérer les réponses et redirections

### 4.1 Retourner une réponse simple



The screenshot shows a code editor with a file named `web.php`. The code defines a route `/response` that returns a simple string:

```
<?php  
use Illuminate\Support\Facades\Route;  
  
Route::get('/response', function () {  
    return response("This is a simple response");  
});  
?>
```

The browser address bar shows `localhost:8000/response`, and the page content is "This is a simple response".

## 4.2 Redirection vers une autre route

The screenshot shows a code editor with a file named `web.php`. The code defines routes for an 'admin' prefix:

```
<?php
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;

Route::prefix('admin')->group(function () {
    Route::get('dashboard', function () {
        return 'Admin Dashboard';
    });

    Route::get('settings', function() {
        return 'Admin Settings';
    });
});

Route::get('/redirect', function () {
    return Redirect('/admin/dashboard');
});
```

To the right, a browser window displays the URL `localhost:8000/admin/dashboard`, showing the page content "Admin Dashboard".

## 4.3 Retourner une réponse JSON

The screenshot shows a code editor with a file named `web.php`. The code defines a route for '/json':

```
<?php
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;

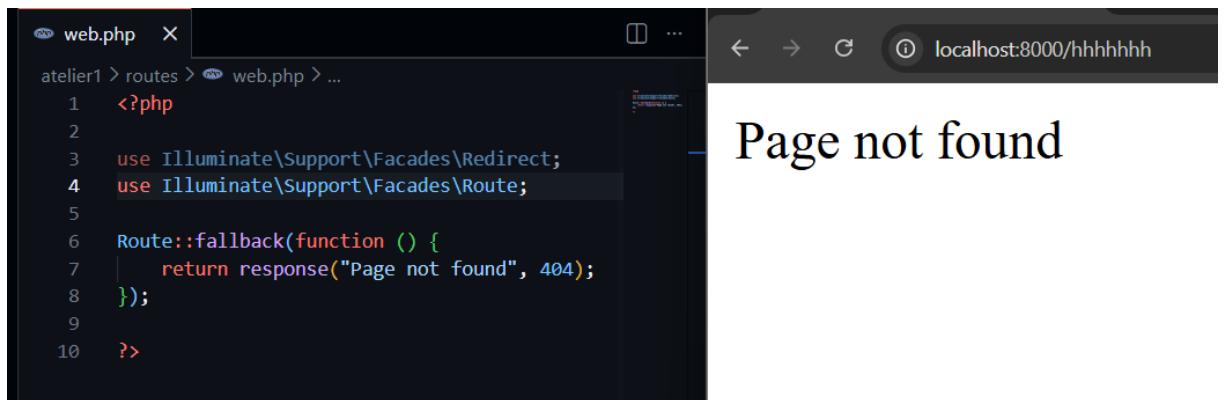
Route::get('/json', function () {
    return response()->json([
        "name" => "Mohamed Said",
        "age" => 20
]);
});
```

To the right, a browser window displays the URL `localhost:8000/json`, showing the JSON response:

```
{ "name": "Mohamed Said", "age": 20 }
```

## 5. Gérer les erreurs 404

### 5.1 Route Fallback

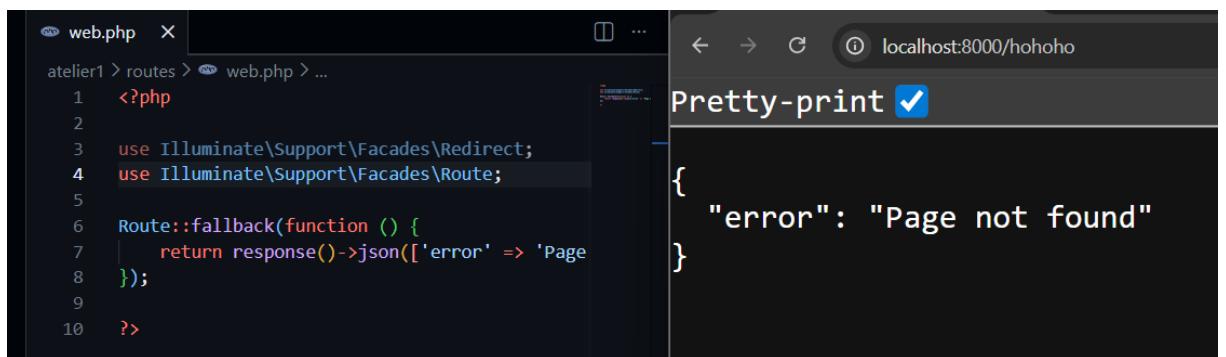


```
<?php
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;

Route::fallback(function () {
    return response("Page not found", 404);
});
```

The screenshot shows a code editor with a file named "web.php". The code defines a route fallback that returns a "Page not found" response with a status code of 404. To the right, a browser window displays the URL "localhost:8000/hhhhhh", resulting in a "Page not found" error message.

### 5.2 Personnalisation des erreurs 404



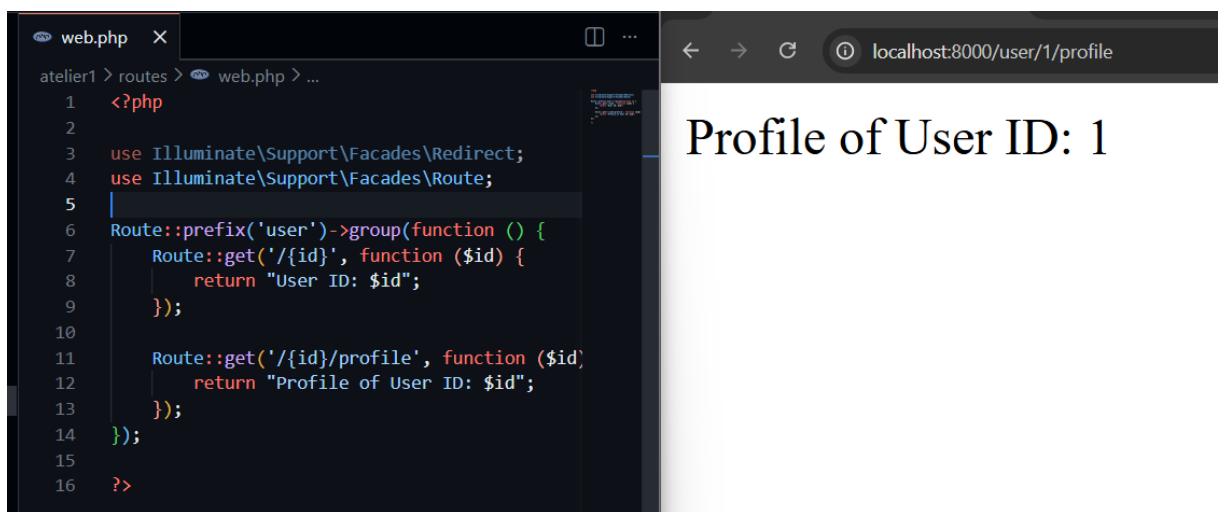
```
<?php
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;

Route::fallback(function () {
    return response()->json(['error' => 'Page not found']);
});
```

The screenshot shows a code editor with a file named "web.php". The code defines a route fallback that returns a JSON response with an "error" key set to "Page not found". To the right, a browser window displays the URL "localhost:8000/hohoho", resulting in a JSON response: {"error": "Page not found"}.

## 6. Bonnes pratiques en matière de routing

### 6.1 Utilisation des groupes de routes



```
<?php
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;

Route::prefix('user')->group(function () {
    Route::get('/{id}', function ($id) {
        return "User ID: $id";
    });

    Route::get('/{id}/profile', function ($id) {
        return "Profile of User ID: $id";
    });
});
```

The screenshot shows a code editor with a file named "web.php". The code defines a route group for the "user" prefix. It contains two routes: one for "/{id}" returning "User ID: \$id" and another for "{id}/profile" returning "Profile of User ID: \$id". To the right, a browser window displays the URL "localhost:8000/user/1/profile", resulting in the response "Profile of User ID: 1".

## 6.2 Routes avec des paramètres optionnels

The image shows a code editor window with a file named `web.php`. The code defines a route:

```
<?php  
use Illuminate\Support\Facades\Redirect;  
use Illuminate\Support\Facades\Route;  
  
Route::get('/user/{id}/{group?}', function ($id, $group = 'default') {  
    return "User ID: $id, Group: $group";  
});  
?>
```

To the right of the code editor is a browser window showing the URL `localhost:8000/user/11/dev203`. The page content is "User ID: 1, Group: dev203".

## 6.3 Routes avec des conditions

The image shows a code editor window with a file named `web.php`. The code defines a route with a condition:

```
<?php  
use Illuminate\Support\Facades\Redirect;  
use Illuminate\Support\Facades\Route;  
  
Route::get('/user/{id}', function ($id) {  
    return "User ID: $id";  
})->where('id', '[0-9]+');  
?>
```

To the right of the code editor is a browser window showing the URL `localhost:8000/user/11`. The page content is "User ID: 11".