

Cours Laravel - Components

Ce support présente le fichier fourni concernant les Components en Laravel. Le code est reproduit ci-dessous avec mise en forme.

```
<?php

// Pour les inclusion de fichier, il est possible d'utiliser une autre approche avec Laravel
// Les components

// @include('components.button')■Pas possible de passer des attributs facilement.
// <x-button color="blue">■ Gestion plus propre des paramètres avec @props.

// Cette approche répond à une syntaxe bien précise.

// https://laravel.com/docs/12.x/blade#components

# php artisan make:component ComponentTest

// cette commande va créer 2 fichiers.
// un fichier contenant la logique PHP dans app/View/Components/ComponentTest.php
// un fichier contenant le code html dans resources/views/components/component-test.blade.php

// il est possible de ne pas générer une classe :

# php artisan make:component mon-component --view

// Laravel convertit automatiquement la syntaxe PascalCase en kebab-case dans les vues
// classe PHP : ComponentTest.php
// vue : component-test.blade.php

// utilisation dans blade pour un include : <x-component-test />

// Lorsque l'on appelle un component dans une vue, Laravel va d'abord chercher la classe dans app/View/Components
// Si aucun n'existe, on obtient une erreur

// Un des avantages c'est de pouvoir passer des attributs dynamiquement
// exemple ici on met une couleur par défaut "blue" mais lors de l'appel on peut donner une valeur d'attribut

// resources/views/components/button.blade.php
@props(['color' => 'blue'])

<button class="px-4 py-2 text-white bg-{{ $color }}-500 rounded">
    {{ $slot }}
</button>

// Dans la vue :
<x-button color="red">Annuler</x-button>

// Slot représente le contenu à insérer

// resources/views/components/card.blade.php
<div class="border p-4 rounded shadow">
    {{ $slot }}
</div>

// Dans la vue
<x-card>
    <p>Ceci est une carte avec du texte dynamique.</p>
</x-card>

// resources/views/components/card.blade.php
@props(['title'])

<div class="border p-4 rounded shadow">
    <h2 class="text-lg font-semibold">{{ $title }}</h2>
    <div>
        {{ $slot }}
    </div>
</div>

// Dans la vue :
<x-card title="Informations">
    <p>Ceci est une carte avec du texte dynamique.</p>
```

```

</x-card>

// <x-slot> pour définir des zones spécifiques
// resources/views/components/card-layout.blade.php
<div class="border p-4 rounded shadow">
    @isset($header)
        <div class="font-semibold text-lg">{{ $header }}</div>
    @endisset

    <div class="mt-2">
        {{ $slot }}
    </div>

    @isset($footer)
        <div class="text-sm text-gray-500 mt-4">{{ $footer }}</div>
    @endisset
</div>

// dashboard.blade.php
<x-card-layout>
    <x-slot name="header">
        Titre de la carte
    </x-slot>

    Contenu principal de la carte.

    <x-slot name="footer">
        Pied de page personnalisé
    </x-slot>
</x-card-layout>

```