



[←Revenir à la théorie.](#)

Les vues dans Laravel

- [1. Définition](#)
- [2. Appel d'une Vue](#)
- [3. Vue paramétrée](#)
 - [3.1 Un paramètre](#)
 - [3.2 Plusieurs paramètres](#)
 - [3.3 Ajout de CSS et de JS](#)

1. Définition

Mais qu'est-ce qu'une vue ?

Il faut prendre cela comme on l'entend: une vue, c'est quelque chose que l'on voit. Et dans le monde du web ce que l'on voit c'est du HTML.

Une vue affichera donc du code HTML et d'autres gentilles choses que nous verrons au fil du temps. 😊

La définition de toutes nos vues se trouve dans le dossier: `resources/views/`

Par défaut, Laravel a une vue par défaut qui s'appelle 'welcome'.

On notera qu'elle se nomme `welcome.blade.php`

Le nom de fichier de nos vues auront donc cette syntaxe: `nomdelavue.php` ou bien `nomdelavue.blade.php`

2. Appel d'une Vue

Nous l'avons déjà vu dans le routage:

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

La route va appeler notre vue 'welcome' dans le répertoire resources/views/

3. Vue paramétrée

3.1 Un paramètre

En général on veut souvent transmettre des données à notre vue. Soit via un contrôleur soit directement via l'url.

Imaginons que nous voulons saluer notre utilisateur.

Notre route:

```
Route::get('/newuser/{user}', function($user){  
    return view('newuser')->with('user',$user);  
});
```

On utilise la méthode with avec comme première paramètre le nom de la variable dans la vue et ensuite la valeur de notre paramètre.

Notre vue newuser.php:

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome </title>
</head>
<body>
  <h1>Bienvenue <?= $user ?> !</h1>
</body>
</html>

```

3.2 Plusieurs paramètres

Dans ce cas-ci au lieu d'utiliser la méthode with on peut envoyer à notre vue un tableau.
Notre route:

```

Route::get('/newuser/{firstname}/{name}', function ($firstname, $name) {
    return view('newuser', ['firstname' => $firstname, 'name' => $name]);
});

```

On pourrait aussi enchaîner deux with:

```

Route::get('/newuser/{firstname}/{name}', function ($firstname, $name) {
    return view('newuser')->with('firstname',$firstname)->with('name',$name);
});

```

Notre vue newuser.php:

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome Nouvel Utilisateur !</title>
</head>
<body>
  <h1>Bienvenue nouvel utilisateur !</h1>
  Ton prénom est <?= $firstname ?> <br/>
  Ton nom est <?= $name ?>
</body>
</html>

```

3.3 Ajout de CSS et de JS

Il suffit de faire référence à nos fichiers css et js qui seront dans le répertoire /public dans notre projet Laravel.

Depuis notre vue, appelons le fichier app.css qui se trouve dans /public/css

```

<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome </title>
  <link rel="stylesheet" ref="/css/app.css">
</head>
<body>
  <h1>Bienvenue nouvel utilisateur !</h1>
  Ton prénom est <?= $firstname ?> <br/>
  Ton nom est <?= $name ?>
</body>
</html>

```

C'est donc dans le répertoire /public que vous allez mettre vos fichiers js et css. Par exemple vos fichiers nécessaires pour Bootstrap.

[←Revenir à la théorie.](#)

[Eqla 2021](#) - Formation Laravel