TWIG

Comment créer des vues dans Symfony ?

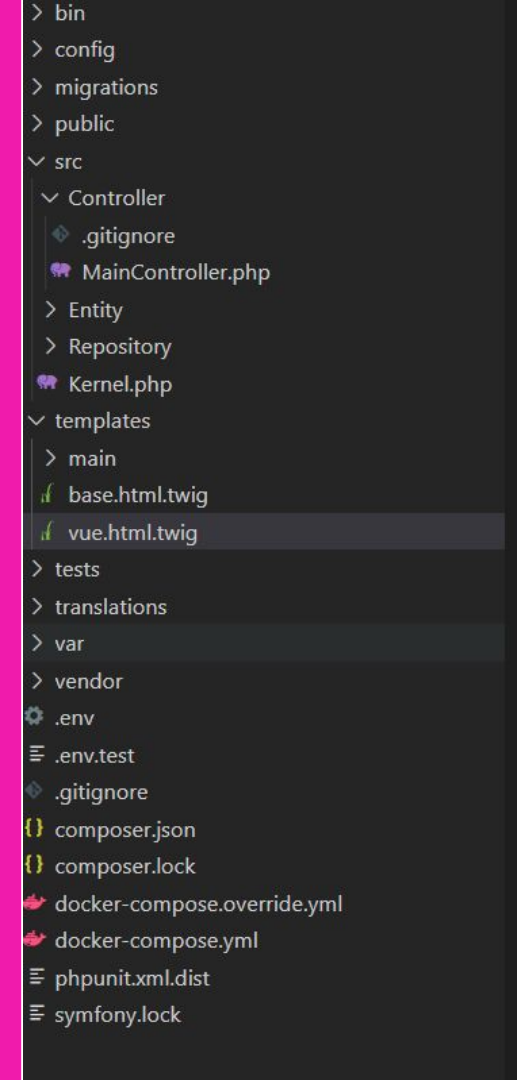
Vue

- Comme vu précédemment les controllers sont les capitaines des projets symfony, ils ne s’occupent que de code PHP.
- Le code HTML sera créé dans les vues des projets.
- Ces vues seront placées dans le dossier templates des projets.
- Les vues seront codées avec TWIG.

TWIG

- Twig est un moteur de templates qui a été créé par SensioLabs, les créateurs de Symfony
- Twig permet ainsi de mélanger du code HTML et du PHP de manière limité.
- Cela permet de créer des vues avec du code HTML et d'y implanter par exemple des variables créer dans les controllers.
- Les fichiers twig auront tous des noms suivant le schéma suivant :
    - nom_du_fichier.html.twig
- Pour appeler un fichier twig dans une méthode dans un controller, il faut utiliser la fonction render() qui vient du AbstractController

TWIG



Emplacement du fichier twig

Code PHP dans le Controller

```php
#[Route('/vue', name: 'vue')]
public function vue()
{
    return $this->render('vue.html.twig');
}
```

Code du fichier twig

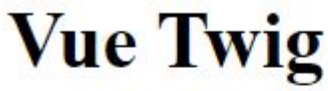
vue.html.twig

templates > vue.html.twig

```twig
<!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>Document</title>
</head>
<body>
    <h1>Vue Twig</h1>
</body>
</html>
```

TWIG

Résultat



TWIG

- Pour intégrer du code PHP dans un fichier twig, il faut utiliser une syntaxe particulière
  - {{ … }} : appel de variables ou de fonctions twig
  - {# … #} : commentaires
  - {% … %} : commande, comme une affectation, une condition, une boucle ou un bloc HTML
- On peut intégrer des variables d’un Controller dans une vue grâce à la fonction render.

TWIG

nom de la variable dans le fichier twig

```php
#[Route('/variable', name: 'variable')]
public function variable()
{

    $variable = "ma Super Variable";

  return  $this->render('variable.html.twig', ['variable' => $variable]);
}
```

Code PHP

attribution de la variable

variable.html.twig

templates > variable.html.twig

```twig
<!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>Document</title>
</head>
<body>
    <h1>{{ variable }}</h1>
</body>
</html>
```

Code Twig

TWIG

Résultat

ma Super Variable

TWIG

```php
#[Route('/tableau', name: 'tableau')]
public function tableau()
{

    $tableau = [
        "article 1",
        "article 2",
        "article 3",
        "article 4"
    ];

  return  $this->render('tableau.html.twig', ['tableau' => $tableau]);
}
```

Code PHP

tableau.html.twig

templates > tableau.html.twig

```twig
<!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>Document</title>
</head>
<body>
    {% for ligne in tableau %}
        <h2>{{ligne}}</h2>
    {% endfor %}
</body>
</html>
```

Code Twig

TWIG

Résultat

---

article 1

article 2

article 3

article 4

TWIG

- Il existe plusieurs fonctions de twig très utile pour le code:
  - path : pour créer des liens vers des routes
  - asset : pour récupérer des fichier ou des dossiers dans le dossier public du projet (image, fichier css, javascript)
  - include : pour inclure du code twig d'un autre fichier
  - extends : pour hériter d'un fichier twig

TWIG

Exemple :

variable.html.twig

templates > variable.html.twig

```twig
<!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.
    <title>Document</title>
</head>
<body>
    <h1>{{ variable }}</h1>
    <a href="{{ path('vue') }}">Page vue</a>
    <img src="{{ asset("/media/BruceBvS.jpg") }}" alt="">
</body>
</html>
```

TWIG

Exemple : Résultat

ma Super Variable

Page vue

TWIG

- Bravo, tu peux maintenant créer des vues grâce à twig sur ton projet.
- En avant matelot !