

Séquence 2



Symfony

OBJECTIFS DE LA SÉQUENCE

» Cette séquence a pour objectifs de

- Créer un contrôleur
- Présenter le moteur TWIG afin de créer des vues



CONTRÔLEUR

URL »»

Une route

http://localhost:8000/bienvenue



Créer un contrôleur



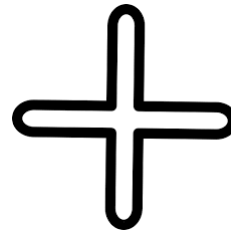
:=> CMD

symfony console make:controller NomDuContrôleur

Example

:=> CMD

symfony console make:controller Bienvenue



CONTRÔLEUR

Route »»



<http://localhost:8000/bienvenue>

```
class BienvenueController extends AbstractController
{
    #[Route('/bienvenue', name: 'app_bienvenue')]
    public function index(): Response
    {
        return $this->render('bienvenue/index.html.twig', [
            'controller_name' => 'BienvenueController',
        ]);
    }
}
```

Une route

Exécution de la
méthode **index**



#Route : annotation ("pseudo-code" permettant de donner des informations de configuration à Symfony)

CONTRÔLEUR

Route »»



[http://localhost:8000](http://localhost:8000/bienvenue) **/bienvenue**



Symfony traite la requête HTTP et extrait la route



Symfony recherche s'il y a un contrôleur qui gère cette route



Si OUI, Symfony :

- ✓ **Instancie la contrôleur**
- ✓ **Exécute la méthode liée à la route**

CONTRÔLEUR

Vue »»



<http://localhost:8000/bienvenue>

```
class BienvenueController extends AbstractController
{
    #[Route('/bienvenue', name: 'app_bienvenue')]
    public function index(): Response
    {
        return $this->render('bienvenue/index.html.twig', [
            'controller_name' => 'BienvenueController',
        ]);
    }
}
```



Appel de la vue



**Retourne une réponse
HTTP contenant le code
HTML de la vue**

Appel de la vue  **TWIG**

Définition »»



- »» Moteur de template pour PHP
- »» Symfony 6 utilise la version **3** de Twig
- »» Permet de séparer le code PHP du code HTML

<https://twig.symfony.com>

Langage »

» **Propose un langage de template utilisant des balises spécifiques**

Trois types de balises

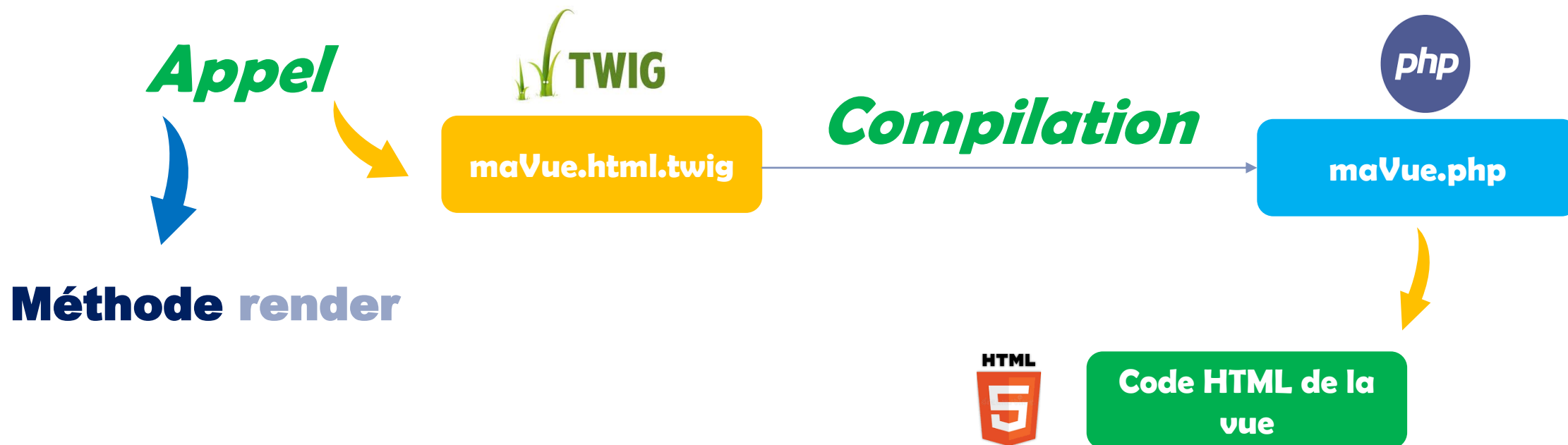
- { % . . . % } : pour exécuter une action
- { # . . . # } : pour définir un commentaire
- {{ . . . }} : pour afficher

» **PLUS DE PHP AU NIVEAU DES VUES !**

TWIG

Principe »»

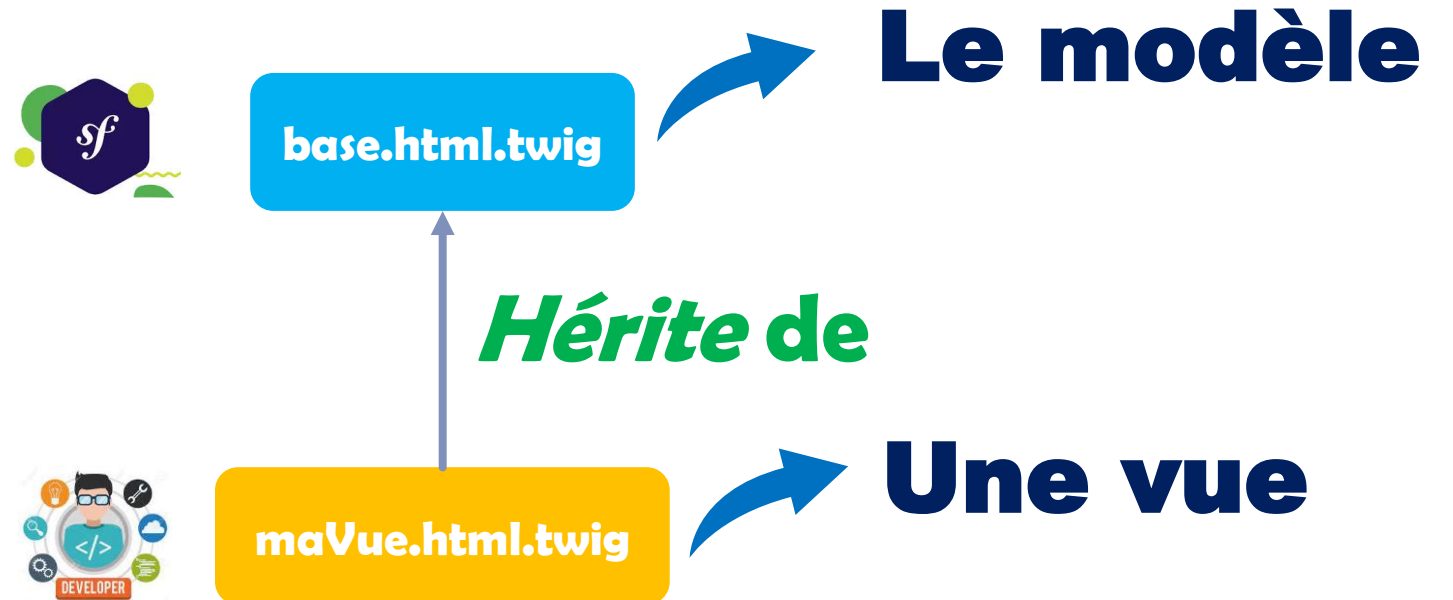
»» Chaque vue est liée à un fichier d'extension **.html.twig**



TWIG

Modèle de vue >>>

>> **Symfony propose un modèle de vue à partir duquel toutes les vues seront créées**



Principe - Bloc »»

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>{% block title %}Welcome!{% endblock %}</title>
    <link rel="icon" href="...">
    {% block stylesheets %}
    {% endblock %}

    {% block javascripts %}
    {% endblock %}
  </head>
  <body>
    {% block body %}{% endblock %}
  </body>
</html>
```

base.html.twig

»» Propose des **blocs**

»» Chaque vue va **utiliser** ce modèle

On dit que la vue **étend** le modèle

»» Chaque vue va **redéfinir** les **blocs** afin d'y insérer son propre code

Redéfinition des blocs »»

Hérite de

```
{% extends 'base.html.twig' %}  
  
{% block title %}Bienvenue{% endblock %}  
  
{% block body %}  
    <p>Bienvenue dans Symfony !</p>  
{% endblock %}
```

index.html.twig

Redéfinition du
bloc title

Redéfinition du
bloc body

»» C'est "COMME SI" on faisait un "COPIER/COLLER" du modèle et que l'on complétait les blocs que l'on souhaite

▶ CONTRÔLEUR

Passage de données à la vue »»



```
return $this->render('vue_Twig', [  
    "cle1" => "valeur1",  
    "cle2" => "valeur2"  
]);
```

```
{% block body %}  
    {{ cle1 }}  
    {{ cle2 }}  
{% endblock %}
```

tableau associatif

CONTRÔLEUR

Route paramétrée >>>



<http://localhost:8000> /bienvenue/**pierre**



<http://localhost:8000> /bienvenue/**paul**

>> Pierre et Paul sont des paramètres de route

>> /bienvenue/prenom est une route paramétrée

CONTRÔLEUR

Route paramétrée >>>



<http://localhost:8000> /bienvenue/**pierre**



<http://localhost:8000> /bienvenue

CONTRÔLEUR

Route paramétrée >>>



<http://localhost:8000/bienvenue/pierre>

```
class BienvenueController extends AbstractController
{
```

```
    #[Route('/bienvenue/{prenom}', name: 'app_bienvenue_prenom')]
    public function bienvenuePrenom(string $prenom): Response
```

```
    {
```

```
        return $this->render('bienvenue/index.html.twig', [
```

```
            'prenom' => $prenom
```

```
        ]);
```

```
    }
```

```
}
```



NEW METHOD

index.html.twig

```
{% extends 'base.html.twig' %}
```

```
{% block title %}Bienvenue{% endblock %}
```

```
{% block body %}
```

```
    <p>Bienvenue à {{ prenom }} dans Symfony !</p>
```

```
{% endblock %}
```

Apprendre par le

