

DAWS – 2º

TWIG con Symfony

JESÚS MOLINA HERNÁNDEZ
jmolina@florida-uni.es

Índice

1. Introducción a Twig
2. Estructura Básica de Twig
3. Renderización de Plantillas
4. Filtros en Twig
5. Herencia e inclusión de plantillas
6. Gestión de Formularios en Twig

1. Introducción a Twig

Twig es un **motor de plantillas** para PHP.

Simplifica la creación de vistas en aplicaciones Symfony.

Ofrece:

- Sintaxis clara y limpia.
- Reutilización de componentes (herencia de plantillas).
- Seguridad integrada.

1. Introducción a Twig

Twig se incluye por defecto en Symfony.

Integración automática con controladores y datos. Método “**render**” en AbstractController.

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

2. Estructura básica de Twig

Symfony siempre nos crea el fichero: “**base.html.twig**” que es la plantilla principal.

Después en el resto de plantillas se extiende la plantilla base y se da contenido a las secciones que se han definido en la base.

→ Verlo en ejemplo de código.

2. Estructura básica de Twig

Sintaxis:

- **Variables:** {{ variable }}
- **Funciones:** {{ función (parámetros) }}
- **Bloques de código:** {% ... %}
 - If, for, else, set, block, extends, include, with.
 - And, or not
 - ...
- **Comentarios:** #{ ... #}

→ Verlo en ejemplo de código.

3. Renderización de plantillas

1. Método en el controlador: **render**

```
return $this->render('ruta/plantilla.html.twig', datos);
```

2. Paso de **datos**

```
return $this->render( view: 'producto/index.html.twig', [
    'productos' => $productos,
    'mensaje' => "Hay $totalProductos productos en total",
]);
            datos
```

3. Renderización en Twig

```
<td>{{ producto.id }}</td>
<td>{{ producto.nombre }}</td>
```

4. Filtros en Twig

Transforman valores de las variables directamente en la plantilla.

Ejemplos:

- Capitalizar: {{ nombre|capitalize }}
- Fecha: {{ fecha|date('d/m/Y') }}
- Recortar: {{ texto|truncate(30) }}

[Enlace a lista completa](#)

5. Herencia e inclusión de plantillas

Son herramientas para reutilizar y organizar mejor el código.

- **Herencia (extend):** Parte de una plantilla existente y le añade más bloques.
- **Inclusión (include):** Permite incrustar una plantilla dentro de otra. Útil para dividir el código y hacerlo más manejable.

6. Gestión de formularios en Twig

Los formularios se crean con Symfony/Form (se verá en la siguiente sesión)
Una vez creado, se integra en las plantillas de Twig mediante:

```
 {{ form_start(form) }}  
   {{ form_widget(form) }}  
   <button class="btn">{{ button_label|default('Save') }}</button>  
 {{ form_end(form) }}
```

Así se renderiza el form por defecto, si se quiere tunear, se puede hacer añadiendo **{{form_row()}}** para los campos deseados

```
 {{ form_start(form) }}  
   {{ form_row(form.nombre, { 'attr': { 'class': 'form-control' } }) }}  
   {{ form_row(form.email, { 'label': 'Tu correo electrónico' }) }}  
 <button class="btn btn-primary">{{ button_label|default('Guardar') }}</button>  
 {{ form_end(form) }}
```