

Fragmentos con parámetros

Curso intermedio de Thymeleaf

Fragmentos con parámetros

- Thymeleaf permite **definir fragmentos como si fueran funciones**, recibiendo un número arbitrario de parámetros.
- El *tipo de dato* puede ser
 - Texto
 - Números
 - Objetos (beans)
 - **Otros fragmentos**
 -

Fragmentos con parámetros

```
<div th:fragment="frag(v1, v2)">  
  <p th:text="${v1}">...</p>  
  <a th:href="@{v2}">...</a>  
</div>
```

```
<div th:replace="::frag(${valor1},${valor2})">...</div>  
<div th:replace="::frag(v1=${valor1}, v2=${valor2})">...</div>
```

Fragmentos con parámetros

```
<div th:fragment="frag(v1)">  
    <p th:text="${v1}">...</p>  
</div>
```

```
<div th:replace="::frag('Hola mundo')">...</div>
```

```
<div>  
    <p>Hola mundo</p>  
</div>
```

Diseños flexibles

Diseños flexibles

- Mecanismo similar al *Thymeleaf Layout Dialect* (estilo jerarquizado).

```
<head th:fragment="common_header(title,links)">
  <title th:replace="${title}">The awesome application</title>
  <!-- Estilos y scripts comunes -->
  <link rel="stylesheet" type="text/css" media="all"
        th:href="@{/css/awesomeapp.css}">
  <link rel="shortcut icon" th:href="@{/images/favicon.ico}">
  <script type="text/javascript"
        th:src="@{/sh/scripts/codebase.js}"></script>
  <!--/* Estilos o elementos adicionales concreta */-->
  <th:block th:replace="${links}" />
</head>
```

Diseños flexibles

- 2 argumentos: *title* y *links*
 - *Title* será utilizado para reemplazar el contenido de la etiqueta *<title>*.
 - Links será reemplazado por el bloque *<th-block>*

```
<head th:fragment="common_header(title,links)">
  <title th:replace="${title}">The awesome application</title>
  ...
  <!--/* Estilos o elementos adicionales concreta */-->
  <th:block th:replace="${links}" />
</head>
```

Diseños flexibles

- Los argumentos pueden ser fragmentos

```
...  
<head th:replace="base :: common_header(~{::title},~{::link})">  
    <title>Awesome - Main</title>  
  
    <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">  
    <link rel="stylesheet"  
        th:href="@{/themes/smoothness/jquery-ui.css}">  
  
</head>  
...
```


Diseños flexibles

- El resultado

...

```
<head>
  <title>Awesome - Main</title>
  <!-- Estilos y scripts comunes -->
  <link rel="stylesheet" type="text/css" media="all"
        href="/awe/css/awesomeapp.css">
  <link rel="shortcut icon" href="/awe/images/favicon.ico">
  <script type="text/javascript" src="/awe/sh/scripts/codebase.js"></script>
  <link rel="stylesheet" href="/awe/css/bootstrap.min.css">
  <link rel="stylesheet" href="/awe/themes/smoothness/jquery-ui.css">
</head>
```

...

Fragmentos vacío y token de no-op

- Podemos utilizar el fragmento vacío `~{}` como argumento:

```
<head th:replace="base :: common_header(~{::title},~{})">  
  <title>Awesome - Main</title>  
</head>
```

- También podemos usar el token no-op (guión bajo) para dejar el valor por defecto.

```
<head th:replace="base :: common_header(_,~{::link})">  
  <title>Awesome - Main</title>  
  <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">  
  <link rel="stylesheet" th:href="@{/themes/smoothness/jquery-ui.css}">  
</head>
```

Eliminación de fragmentos

- Sirve, sobre todo, para eliminar código de prototipado, o en evaluaciones de expresiones complejas.
- Atributo **th:remove**
 - all
 - body
 - tag
 - all-but-first
 - none

Herencia de *layout*

Herencia de layout

- Podemos jugar con fragmentos con parámetros para tener una plantilla base.
- Nos permite no tener que usar el *Thymeleaf Layout Dialect*.
- Definir fragmentos a nivel de `<html>`

Herencia de layout

```
<!DOCTYPE html>
<html th:fragment="layout(titulo, contenido)"
xmlns:th="http://www.thymeleaf.org">
<head>
    <title th:replace="${titulo}">Título</title>
</head>
<body>
    <h1>Layout H1</h1>
    <div th:replace="${contenido}">
        <p>Contenido del layout</p>
    </div>
    <footer>
        Pie del layout
    </footer>
</body>
</html>
```

Reto

- Refactorizar el resto de plantillas (en la carpeta admin) para que utilicen nuestra plantilla base.
- Añadir un nuevo parámetro al fragmento base, que nos permita destacar el enlace del menú como activo para la página en la que nos encontramos.