Desarrollo web en entorno servidor

UT1.2 – Programación web

8 – Thymeleaf – Utilidades – Variables

IES Clara del Rey – Madrid

Las expresiones {#utilidad} y {#variable}

Dentro de las expresiones ${...} y @{...} se puede usar:

- Expresiones de clases de utilidad
- Expresiones de variables

Las dos se construyen con #, y el nombre de la clase de utilidad o la variable que se desea utilizar. Ejemplos:

- Usar clase de utilidad para comprobar si una colección está vacía:

```
<span th:if="${#lists.isEmpty(datos)}">No hay datos</span>
```

- Usar una variable para mostrar el locale (idioma + cultura) actual

```
<p>Idioma: <span th:remove="tags" th:text="${#locale.getDisplayName()}" >
                                                Nombre del idioma</span></p>
```

Las expresiones {#utilidad} y {#variable}

No hay que confundir las expresiones de utilidad o de variable con las expresiones de mensajes:

- Expresiones de utilidad o variable, se usan dentro de otras expresiones: ${#utilidad.método(...)}, ${#variable} – El símbolo "#" acompaña a la clase de utilidad, o a la variable.
- Expresiones de mensajes, que pueden contener otras expresiones: #{clave_del_mensaje} – El símbolo "#" se escribe fuera de las llaves.

Las expresiones de utilidad o de variable no se pueden usar:

- Dentro de expresiones de selección – *{...} que se usan con th:object.
- Dentro de expresiones de mensajes para localización – #{...}.

Funciones de utilidad

- #dates – Trabajar con fechas. Date, y otras clases relacionadas con fechas, del paquete java.util.
- #temporals – Trabajar con fechas. LocalDate, LocalDatetime, y otras clases relacionadas con fecha y hora, del paquete java.time.
- #calendars – Trabajar con calendarios java.util.Calendar
- #strings, #numbers, #bools – Cadenas, números y booleans.
- #collections, #arrays, #lists, #sets, #maps, #objects – Trabajar con arrays, colecciones y objetos
- Otras: #ids, #uris, #objects, #aggregates, #messages, y más…

Referencia oficial del paquete org.thymeleaf.expression

Variables de utilidad

- #templateName – Nombre de la plantilla actual
- #ctx – Contexto de ejecución de la plantilla.
- #vars – Variables del contexto.
- #locale – La cultura (idioma) en la que se ejecuta la plantilla.
- #request – Sólo en web – La petición (objeto HttpServletRequest).
- #response – Sólo en web – La respuesta (objeto HttpServletResponse).
- #session – Solo en web – La sesión (objeto HttpSession).
- #statics – Permite acceder a atributos estáticos de clases.

Funciones y variables de utilidad en Spring

El dialecto SpringStandard añade algunas cosas a las variables y funciones de utilidad. Esto permite una mayor integración de las plantillas con Spring. Algunas de estas adiciones:

- #environment – Entorno y configuración de Spring.
- #springBean – Acceso al contenedor IoC de Spring.
- #applicationContext – Similar a #springBean, pero permite obtener beans a partir del contexto de la aplicación.
- #authentication, #authorization, #userPrincipal #principal – Relacionadas con la seguridad en la aplicación Spring.

Y alguna otra más.

Ejemplos

- Dar formato a una fecha de tipo java.util.Date:

```
<p th:text="${#dates.format(fechaNacimiento, 'dd/MM/yyyy')}"></p>
```

- Dar formato a una fecha de tipo java.time.LocalDate:

```
<p th:text="${#temporals.format(fechaNacimiento, 'dd/MM/yyyy')}"></p>
```

- Formatear número con dos decimales:

```
<p th:text="${#numbers.formatDecimal(importe, 2)}"></p>
```

- Convertir cadena a mayúsculas:

```
<p th:text="${#strings.toUpperCase('Se puede usar cadenas directamente')}"></p>
```

- Obtener el tamaño de un array:

```
<p th:text="${#arrays.size(myArray)}"></p>
```

Ejemplos

- Filtrar una lista para mostrar solo los elementos mayores de 10:

  *<p th:text="${#collections.filter(numeros, x -> x > 10)}">Lista en forma [a, b, ...] *

- Filtrar, pero para usarlo en th:each:

  *<li th:each="item : ${#collections.filter(numeros, x -> x > 10)}" th:text="${item}">
*

- Obtener el valor de un Map por su clave:

  * *

- Comprobar si una lista contiene un elemento concreto:

  * *

- Mostrar la cultura en la forma xx_XX (ej. es_ES):

  * *