

SINTAXIS BÁSICA

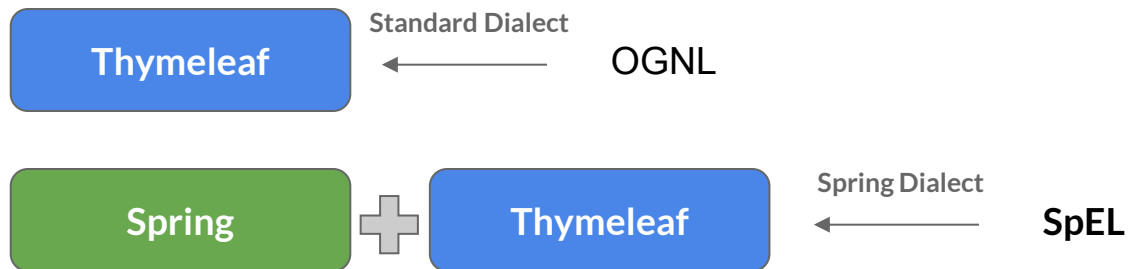


OGNL: Object-Graph Navigation Language

- ▶ Lenguaje de expresiones para Java
- ▶ Permite tomar y establecer propiedades
- ▶ También permite llamada a métodos.
- ▶ Puede servir para manejar arrays
- ▶ Utilizado entre otros, por Thymeleaf.

SpEL: Spring Expression Language

- ▶ Lenguaje de expresiones para Spring
- ▶ Funcionalidades casi idénticas que OGNL.
- ▶ Puede ser usado con los diferentes módulos y tecnologías de Spring.



EXPRESIONES BÁSICAS

- ▶ Expresiones variables: `${...}`
- ▶ Expresiones variables de selección: `*{...}`
- ▶ Expresiones de mensaje: `#{...}`
- ▶ Expresiones de enlaces: `@{...}`
- ▶ Expresiones de fragmentos: `~{...}`

EXPRESIONES LITERALES

- ▶ Textuales: 'texto', 'otro texto', ..
- ▶ Numéricos: 0, 34, 14.2, 3.0, ...
- ▶ Booleanos: true, false
- ▶ Nulos: null
- ▶ ...



1.

EXPRESIONES VARIABLES

EXPRESIONES VARIABLES

- ▶ `${...}`
- ▶ Thymeleaf busca en el contexto una variable con el nombre entre llaves.

```
graph LR; controlador --> plantilla;
```

model.addAttribute("today", new Date()); \${today}

controlador plantilla

- ▶ También puede ser la llamada a un método *getter* como propiedad

```
model.addAttribute("producto", producto);           ${producto.nombre}
```

EXPRESIONES **VARIABLES**

- ▶ Acceso a propiedades usando []

`${persona.padre.nombre} ≈`
`${persona['padre']['nombre']}`

- ▶ Si el objeto es de tipo *map*, podemos usar notación de punto o corchetes.

`${personas['Luismi'].edad} ≈ ${personas.Luismi.edad}`

- ▶ Acceso en base a índice en arrays

`${arrayPersonas.[0].nombre}`

- ▶ Llamadas a métodos

`${persona.nombreCompleto()}`



2.

EXPRESIONES DE SELECCIÓN

EXPRESIONES DE SELECCIÓN

- ▶ Las expresiones de selección o expresiones en selección son también expresiones variables.
- ▶ Diferencia: se ejecutan en el marco de un objeto seleccionado
- ▶ `*{...}`

