



2.

VALIDACIÓN DE DATOS CON SPRING

VALIDACIÓN CON **SPRING**

- ▶ Spring permite usar el estándar *JSR-303/JSR-380 Bean Validation API*.
- ▶ Spring Boot configura por defecto la implementación de este estándar realizada por *hibernate*.
- ▶ Permite realizar la validación añadiendo anotaciones en nuestras clases modelo.

ANOTACIONES DE VALIDACIÓN

- ▶ @NotNull: el atributo no puede ser nulo
- ▶ @Min, @Max: mayor o igual (o menor o igual) que un valor determinado.
- ▶ @NotEmpty: el atributo no puede estar vacío (Strings, colecciones, arrays,...)
- ▶ @Email: el atributo debe ser un email válido.
- ▶ @Size: el atributo tiene que tener un tamaño según el indicado.

<https://beanvalidation.org/2.0/spec/#builtinconstraints>

CUSTOMIZACIÓN DE MENSAJES DE VALIDACIÓN

- ▶ Podemos modificar el mensaje por defecto.
- ▶ Varias estrategias
 - ▷ Cadena *hardcodeada* en la anotación.
 - ▷ **Ficheros de properties.**
 - ▷ Necesitamos configuración algunos beans
 - ▷ Nos permite internacionalización de mensajes de error.

CUSTOMIZACIÓN DE MENSAJES DE VALIDACIÓN

► Configuración

```
@Configuration  
public class MyConfig {
```

```
    @Bean  
    public MessageSource messageSource() {  
        ReloadableResourceBundleMessageSource messageSource  
            = new ReloadableResourceBundleMessageSource();  
  
        messageSource.setBasename("classpath:errors");  
        messageSource.setDefaultEncoding("UTF-8");  
        return messageSource;  
    }
```

```
    @Bean  
    public LocalValidatorFactoryBean getValidator() {  
        LocalValidatorFactoryBean bean = new LocalValidatorFactoryBean();  
        bean.setValidationMessageSource(messageSource());  
        return bean;  
    }
```

Indicamos dónde debe buscar el fichero de properties.

Indicamos que use ese fichero para la validación.

```
}
```

CUSTOMIZACIÓN DE MENSAJES DE VALIDACIÓN

- Cambio de valor

```
public class Empleado {  
  
    @Min(value=1, message="{empleado.id.mayorquecero}")  
    private long id;
```

- Nuevo mensaje de error

```
empleado.id.mayorquecero=El ID del empleado debe ser un numero entero positivo
```

FORMULARIOS CON THYMELEAF

Accede a nuestro [Curso de Introducción a Thymeleaf](#) y aprende cómo utilizar otros campos de formulario más complejos.

Curso de Introducción a Thymeleaf

Aprende a usar Thymeleaf, el mejor motor de plantillas para utilizar junto a Spring y empieza a disfrutar de las ventajas del natural templating con este curso.

★★★★★ 4.4 (9 valoraciones) 4 horas y 8 minutos

CONOCIMIENTOS Y HABILIDADES QUE ADQUIERES REALIZANDO ESTE CURSO

- 🔗 Aprenderás a crear plantillas web.
- 🔗 Conocerás cómo configurar aplicaciones web con Spring Boot, Spring Data JPA y Thymeleaf.
- 🔗 Serás capaz de manejar formularios.
- 🔗 Aprenderás a manejar listas y bucles para representar datos.
- 🔗 Serás capaz de cambiar los estilos CSS de forma condicional.
- 🔗 Aplicarás técnicas de validación y manejo de mensajes de error.

ALGO PARA PRACTICAR

Un ejercicio para hacer por tu cuenta

VALIDACIÓN DE OTROS CAMPOS DEL FORMULARIO

- ▶ Siguiendo el esquema de esta lección, añade el código necesario para visualizar los errores de validación de los demás campos, añadiendo un mensaje personalizado de error.