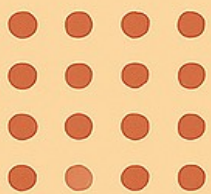


SPRING BOOT



THYMELEAF DATA VALIDATION



Kerin Aguilera

1. Proyecto Springboot themelectf

creamos un saludo controlador que tendra todo

```
package Halo.unsc.Halo;

import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

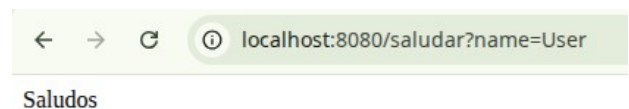
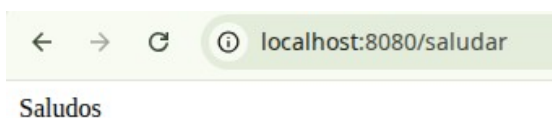
@RestController
public class saludosControlador {
    @GetMapping ("/saludar")
    public String saludar(@RequestParam(name = "nombre", required = false, defaultValue = "Mundo") String nombre, Model modelo) {
        modelo.addAttribute("name", nombre);
        return "Saludos";
    }
}
```

luego hacemos un html que devuelve un saludo con nombre

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Saludar en Springboot</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<p th:text="|nombre, ${name}!!" />
</body>
</html>
```

El saludo por defecto

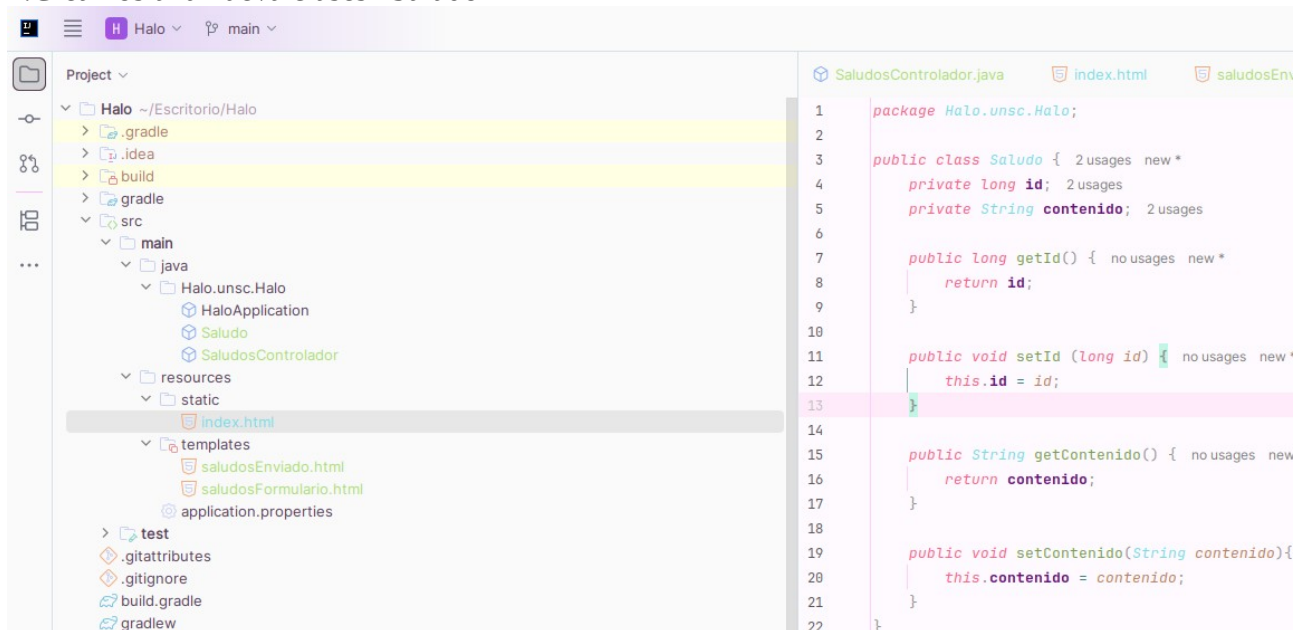
Aquí debe existir un saludo con nombre pero no salio



1. Creamos un controlador.

```
1 package halo.unsc.halo,
2
3 import org.springframework.ui.Model;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.ModelAttribute;
6 import org.springframework.web.bind.annotation.PostMapping;
7 import org.springframework.web.bind.annotation.RestController;
8
9 @RestController no usages new *
10 public class SaludosControlador {
11     @GetMapping("/saludos") no usages new *
12     @ public String saludosFormulario(Model modelo) {
13         modelo.addAttribute( attributeName: "saludo", new Saludo());
14         return "saludo, soy el Saludos formulario";
15     }
16     @PostMapping("/saludos") no usages new *
17     @ public String saludosEnviado(@ModelAttribute Saludo saludo, Model modelo) {
18         modelo.addAttribute( attributeName: "saludo", saludo);
19         return "resultado";
20     }
21 }
22
```

2. Creamos una nueva clases “Saludo”



The screenshot shows an IDE with a project named 'Halo' located at '~/Escritorio/Halo'. The project structure is as follows:

- Project Halo
 - .gradle
 - .idea
 - build
 - gradle
 - src
 - main
 - java
 - halo.unsc.halo
 - HaloApplication
 - Saludo
 - SaludosControlador
 - resources
 - static
 - index.html
 - templates
 - saludosEnviado.html
 - saludosFormulario.html
 - test
 - .gitattributes
 - .gitignore
 - build.gradle
 - gradlew

The right pane shows the code for `SaludosControlador.java`:

```
1 package halo.unsc.halo;
2
3 public class Saludo { 2 usages new *
4     private long id; 2 usages
5     private String contenido; 2 usages
6
7     public long getId() { no usages new *
8         return id;
9     }
10
11     public void setId (long id) { no usages new *
12         this.id = id;
13     }
14
15     public String getContenido() { no usages new *
16         return contenido;
17     }
18
19     public void setContenido(String contenido){
20         this.contenido = contenido;
21     }
22 }
```

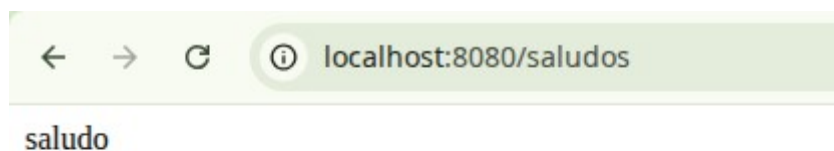

3. creamos un formulario.

```
1 <!DOCTYPE HTML>
2 <html xmlns:th="https://www.thymeleaf.org"> URI is not registered (Settings | Languages & Fra
3 <head>
4     <title>Somos el saludo!!!</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 </head>
7 <body>
8 <h1>Form</h1>
9 <form action="#" th:action="/saludos" th:object="${saludo}" method="post">
10     <p>Id: <input type="text" th:field="*{id}" /></p>
11     <p>mensaje: <input type="text" th:field="*{contenido}" /></p>
12     <p><input type="submit" value="Submit" /> <input type="reset" value="Reset" /></p>
13 </form>
14 </body>
15 </html>
```

4. creamos el resultado del formulario.

```
1 <!DOCTYPE HTML>
2 <html xmlns:th="https://www.thymeleaf.org"> URI is not registered (Settings
3 <head>
4     <title>Soy el resultado</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 </head>
7 <body>
8 <h1>Resultado</h1>
9 <div th:text="'id: ' + ${saludo.id}" />
10 <p th:text="'contenido: ' + ${saludo.contenido}" />
11 <a href="/saludos">Enviamos de nuevo</a>
12 </body>
13 </html>
```

Problemas.



El problema que surge que aunque si, nos da un saludo, no es lo que necesita, sino un formulario y el resultado de se formulario, averiguare que fallo es y luego lo hago

1. Creamos una Clase persona Formulario.

```
4      import jakarta.validation.constraints.NotNull;
5      import jakarta.validation.constraints.Size;
6
7      public class PersonaFormulario { 2 usages
8          @NotNull 3 usages
9          @Size(min =2 , max = 30)
10         private String nombre;
11
12         @NotNull 3 usages
13         @Min(18)
14         private Integer edad;
15         public String getNombre(){ no usages
16             return this.nombre;
17         }
18         public void setNombre(String nombre){ no usages
19             this.nombre = nombre;
20         }
21
22         public Integer getEdad(){ no usages
23             return this.edad;
24         }
25
26         public void setEdad(Integer edad){ no usages
27             this.edad = edad;
28         }
29     }
```

Aqui es donde comprobaremos si las edades son validad o no

2. Creamos el controlador.

```
1 package com.example.formulario;
2
3 import jakarta.validation.Valid;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.validation.BindingResult;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.PostMapping;
8 import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
9 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
10
11 @Controller no usages
12 public class ControladorWeb implements WebMvcConfigurer {
13     @Override 2 usages
14     public void addViewControllers(ViewControllerRegistry registry) {
15         registry.addViewController("/resultados").setViewName("resultados");
16     }
17
18     @GetMapping("/") no usages
19     public String verFormulario(PersonaFormulario personaFormulario){
20         return "formulario";
21     }
22
23     @PostMapping("/") no usages
24     public String checkPersonaInfo(@Valid PersonaFormulario personaFormulario, BindingResult bindingResult) {
25         if (bindingResult.hasErrors()) {
26             return "formulario";
27         }
28     }
29 }
```

3. Creamos la plantilla del formulario.

```
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org"> URI is not registered (Settings | Languages & Frameworks | Schem
3 <body>
4 <form action="#" th:action="@{/}" th:object="${personaFormulario}" method="post">
5     <table>
6         <tr>
7             <td>Name: </td>
8             <td><input type="text" th:field="*{nombre}" /></td>
9             <td th:if="${#fields.hasErrors('nombre')}" th:errors="*{nombre}>Error, mal nombre</td>
10         </tr>
11         <tr>
12             <td>Age: </td>
13             <td><input type="text" th:field="*{edad}" /></td>
14             <td th:if="${#fields.hasErrors('edad')}" th:errors="*{edad}>Error en la edad</td>
15         </tr>
16         <tr>
17             <td><button type="submit">Enviar</button></td>
18         </tr>
19     </table>
20 </form>
21 </body>
22 </html>
```

4. Creamos la repuesta del cuerpo

```
1 <html> Missing required 'lang' attribute.  
2 <body>  
3 Felicidades! tu tienes la edad para registrarte en este sitio!!!.  
4 </body>  
5 </html>
```

5. Los resultados correctos.

← → ↻ ⓘ localhost:8080

Name:

Age:

← → ↻ ⓘ localhost:8080/resultados

Felicidades! tu tienes la edad para registrarte en este sitio!!!.

6. Los resultados y introducimos campos vacios.

← → ↻ ⓘ localhost:8080

Name: el tamaño debe estar entre 2 y 30

Age: no debe ser nulo

7. Si es menor edad

Name:

Age: debe ser mayor que o igual a 18