

PAPW

Alberto Benavides

Ago - Dic 2018

18. Controladores

Crear un controlador

```
// src/main/java/com/example/UserController.java

package com.example;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import // continuar con la siguiente línea
org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class UserController {
    // Funciones
}
```

Métodos de petición y retorno

- `@GetMapping("/ejemplo")` : Atiende una petición `GET` a la URL `/ejemplo`
- `@PostMapping("/ejemplo")` : Atiende una petición `POST` a la URL `/ejemplo`
- `return "ejemplo"` : Regresa al cliente la página `ejemplo.html`

Método para ver un usuario por su id

```
// UserController.java
@GetMapping("/user/{id}")
public String userId(
    Model model,
    @PathVariable(value="id") final Integer id)
    throws URISyntaxException, SQLException {

    UserJDBCTemplate userTemplate = new UserJDBCTemplate();
    userTemplate.setDataSource(Main.getConnection());
    User user = userTemplate.getUser(id);
    model.addAttribute("user", user);

    return "user";
}
```

Método para crear un usuario

```
// UserController.java
@GetMapping("/setUser")
public String userForm(Model model) {
    model.addAttribute("user", new User());
    return "setUser";
}

@PostMapping("/user")
public String userSubmit(@ModelAttribute User user)
    throws URISyntaxException, SQLException {

    UserJdbcTemplate userTemplate = new UserJdbcTemplate();
    userTemplate.setDataSource(Main.getConnection());
    userTemplate.create(user);

    String url = "/user/" + user.getId();
    return url;
}
```

Método para editar un usuario

```
@PostMapping("/userEdit")
public String userEditSubmit(@ModelAttribute User user)
    throws URISyntaxException, SQLException {

    UserJDBCTemplate userTemplate = new UserJDBCTemplate();
    userTemplate.setDataSource(Main.getConnection());
    userTemplate.update(
        user.getId(),
        user.getName(),
        user.getAge()
    );

    return "updateUser";
}
```

Vista para ver un usuario por `id`

```
<!-- src/main/resources/templates/user.html -->
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>
      Perfil de <span th:text = "${user.name}"/>
    </title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8" />
  </head>
  <body>
    <h1>Página de perfil</h1>
    <p th:text="'Nombre: ' + ${user.name}" />
    <p th:text="'Edad: ' + ${user.age}" />
    <a href="/setUser">Crear nuevo usuario</a>
  </body>
</html>
```


Vista para registrar un usuario

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Nuevo usuario</title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8" />
  </head>
  <body>
    <form action="#" th:action="@{/setUser}"
      th:object="${user}" method="post">
      <p>Nombre:
        <input type="text" th:field="*{name}" /></p>
      <p>Edad:
        <input type="number" th:field="*{age}" /></p>
      <p>
        <input type="submit" value="Registrar" /></p>
    </form>
  </body>
</html>
```

Vista para editar un usuario

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8" />
  </head>
  <body>
    <form action="#" th:action="@{/updateUser}"
      th:object="${user}" method="post">
      <input type="hidden"
        th:field="*{id}" value="${user.id}"/>
      <p>Nombre:
        <input type="text" th:field="*{name}"
          value="${user.name}"/></p>
      <p>Edad: <input type="number"
        th:field="*{age}" value="${user.age}"/></p>
      <p><input type="submit" value="Actualizar" /> </p>
    </form>
  </body>
</html>
```

Fuentes

- <https://spring.io/guides/gs/handling-form-submission/>
- <https://stackoverflow.com/questions/7021084/how-do-you-receive-a-url-parameter-with-a-spring-controller-mapping>