

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



TÀI LIỆU THỰC HÀNH
SPRING BOOT

CHƯƠNG 03: FORM DỮ LIỆU SPRING BOOT

Giảng viên biên soạn: Ths. Nguyễn Minh Tân

2024

BUỔI 3: FORM DỮ LIỆU SPRING BOOT

I. MỤC TIÊU

- Khởi tạo trang Web đầu tiên
- Khởi tạo trang web hiển thị thông tin sinh viên
- Tạo Templates Thymeleaf
- Spring Boot Validation
- Controller
- Bootstrap 5.3.0

II. NỘI DUNG THỰC HÀNH:

Khởi tạo trang web hiển thị tên sinh viên

Xây dựng form nhập thông tin sinh viên, có chứa các kiểm tra ràng buộc sau:

- Tên bắt buộc nhập.
- Tuổi phải là Số, có độ giá trị độ tuổi từ 18 đến 100.
- Khoa phải là Chữ

Sau khi nhập thông tin sinh viên. Trang sẽ trả về kết quả thông tin sinh viên.

Để xây dựng một form nhập thông tin sinh viên với các ràng buộc kiểm tra trên ta có thể sử dụng Spring Boot, Thymeleaf cùng với Spring Validation. Dưới đây là các bước để thực hiện điều này:

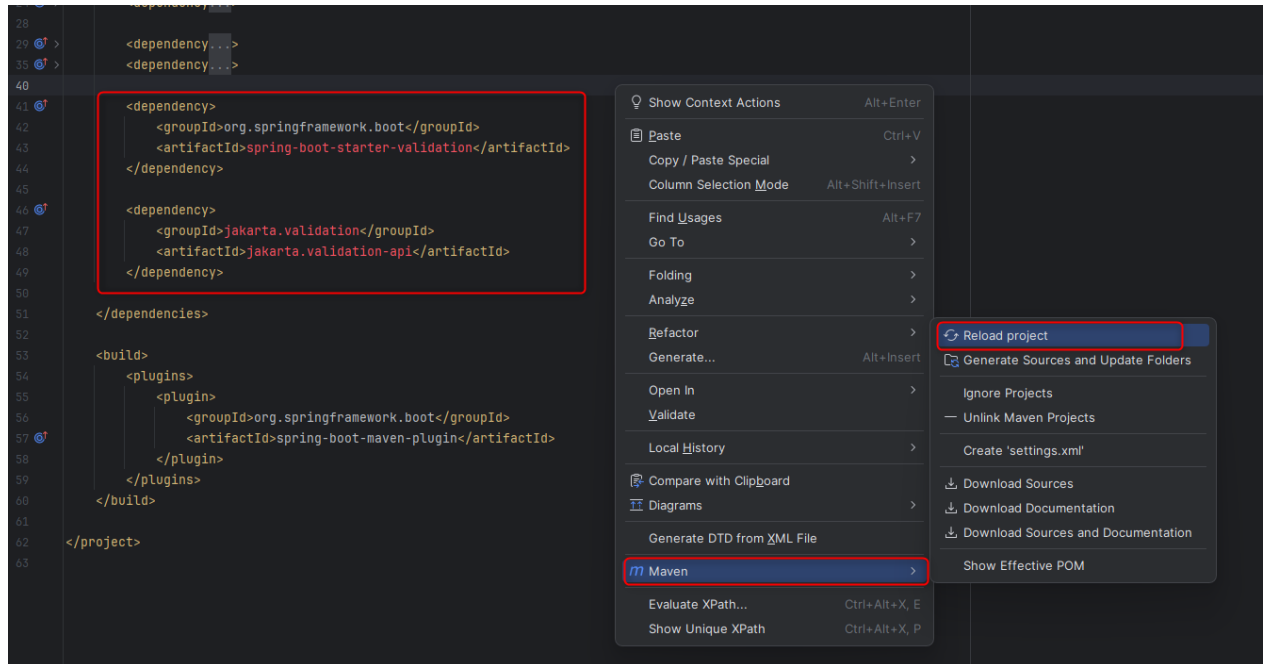
Bước 1: Thêm Dependencies vào pom.xml

Thêm **Dependency Spring Boot Validation (Spring Boot Starter Validation)** và **Jakarta Validation** vào trong file **pom.xml** và trong thẻ **dependencies**.

```
<dependencies>
<!--Các Dependency khác-->
<!--Thêm phần Dependency ràng buộc-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
<dependency>
    <groupId>jakarta.validation</groupId>
    <artifactId>jakarta.validation-api</artifactId>
</dependency>
```

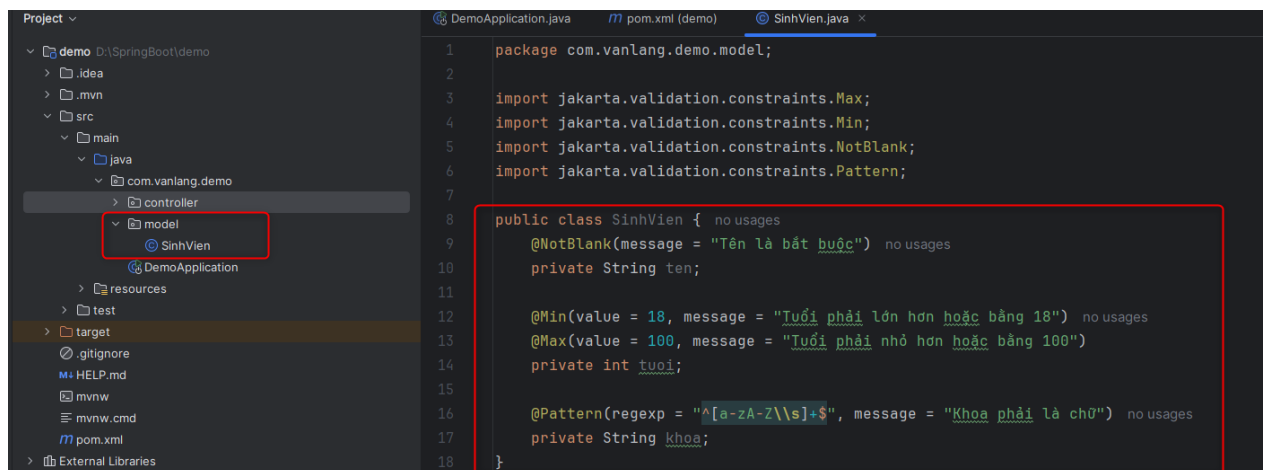
```
</dependencies>
```

Sau khi thêm **Dependency** ta tiến hành chuột phải chọn **Maven** -> chọn **Reload project** để dự án tải và cập nhật **Spring Boot Validation** vào dự án.

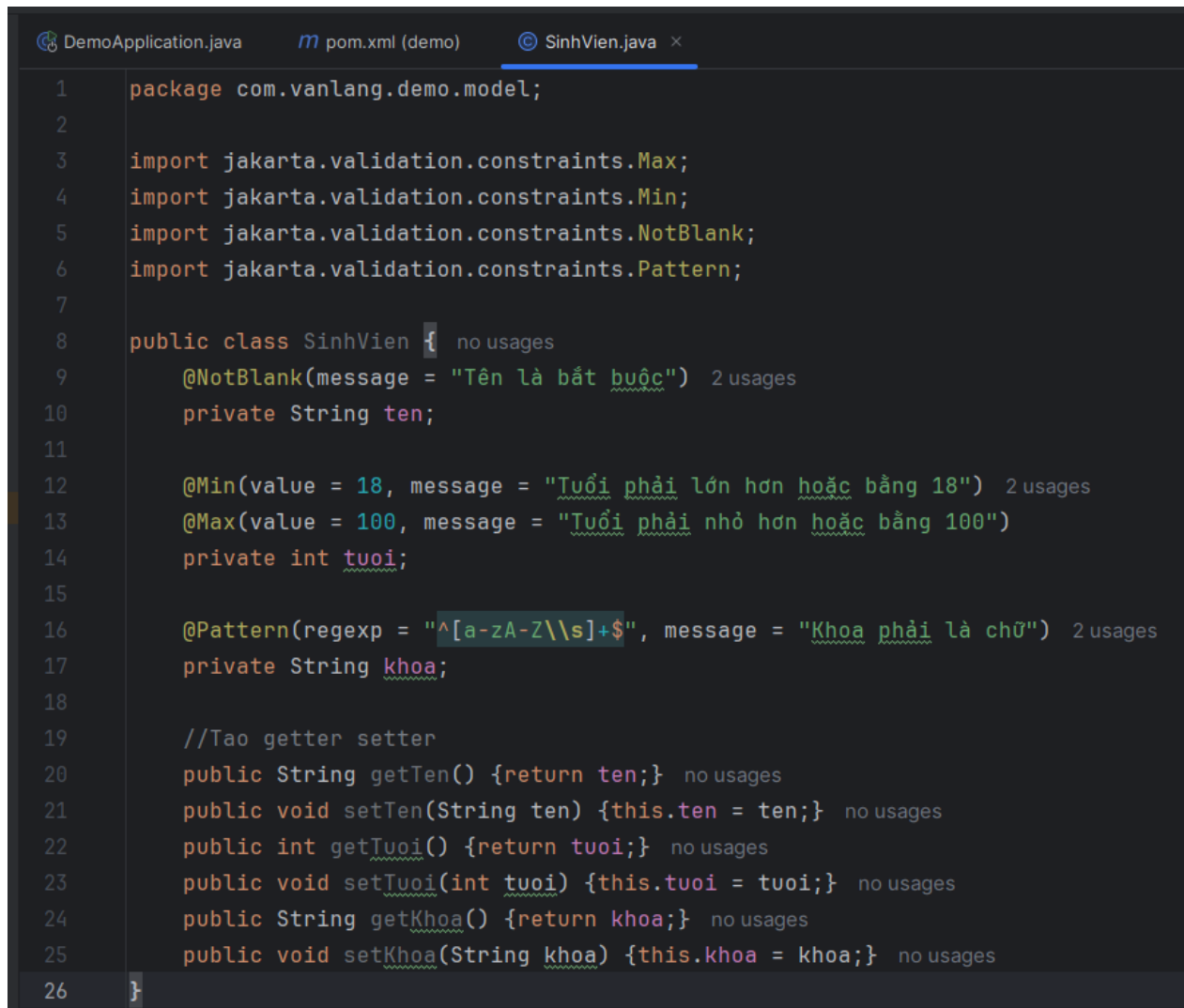


Bước 2: Tạo package model và xác định class SinhVien.java:

Tạo package model tại `src/main/java/com.vanlang.demo`. Nhấn chuột phải vào `com.vanlang.demo`, chọn **New** → **Package** → đặt tên là **model**. Trong package model class **SinhVien.java** với các Validation phù hợp:



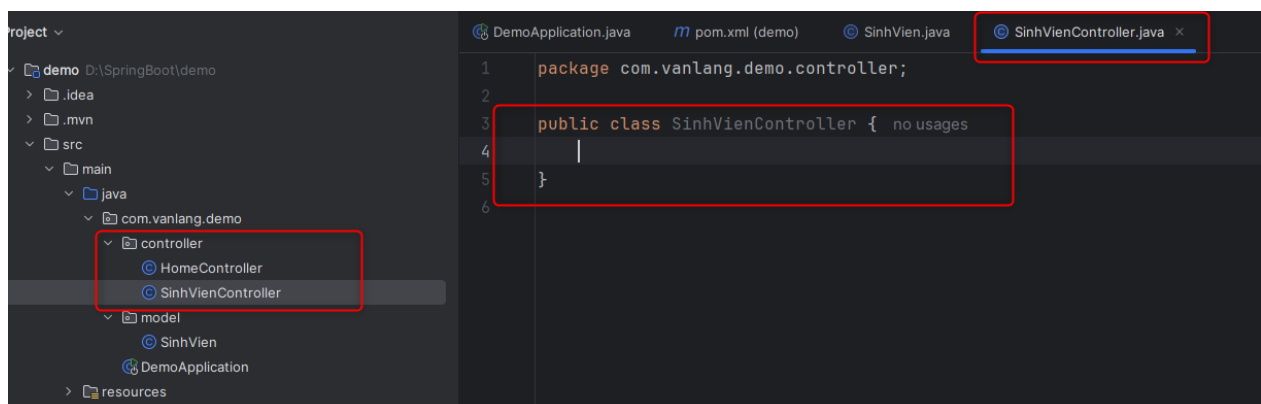
Code mẫu class `SinhVien.java`



```
1 package com.vanlang.demo.model;
2
3 import jakarta.validation.constraints.Max;
4 import jakarta.validation.constraints.Min;
5 import jakarta.validation.constraints.NotBlank;
6 import jakarta.validation.constraints.Pattern;
7
8 public class SinhVien { no usages
9     @NotBlank(message = "Tên là bắt buộc") 2 usages
10     private String ten;
11
12     @Min(value = 18, message = "Tuổi phải lớn hơn hoặc bằng 18") 2 usages
13     @Max(value = 100, message = "Tuổi phải nhỏ hơn hoặc bằng 100")
14     private int tuoi;
15
16     @Pattern(regexp = "[a-zA-Z\\s]+$", message = "Khoa phải là chữ") 2 usages
17     private String khoa;
18
19     //Tao getter setter
20     public String getTen() {return ten;} no usages
21     public void setTen(String ten) {this.ten = ten;} no usages
22     public int getTuoi() {return tuoi;} no usages
23     public void setTuoi(int tuoi) {this.tuoi = tuoi;} no usages
24     public String getKhoa() {return khoa;} no usages
25     public void setKhoa(String khoa) {this.khoa = khoa;} no usages
26 }
```

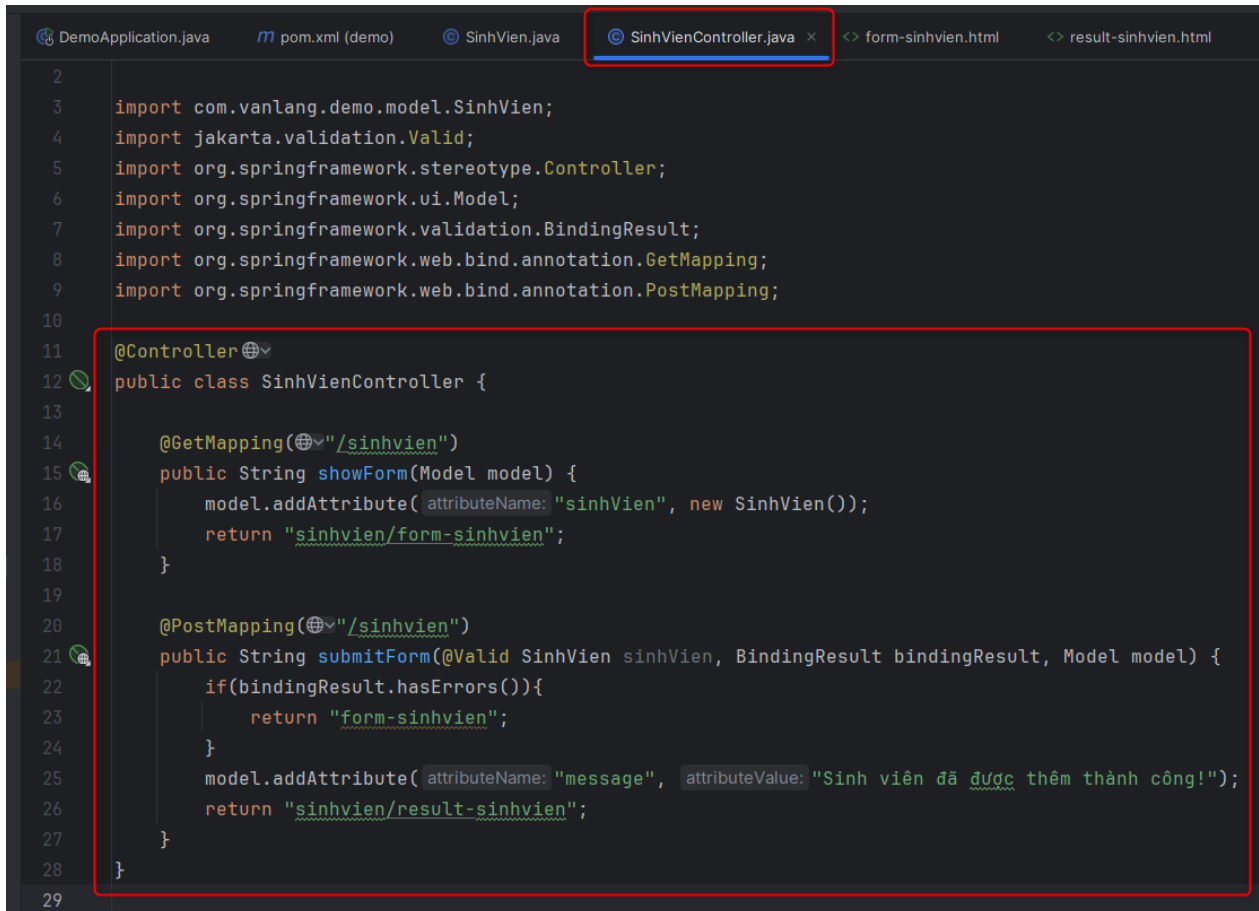
Bước 3: Thêm Controller

Tạo thêm **SinhVienController.java** để xử lý các yêu cầu GET và POST, bao gồm Validation.



```
1 package com.vanlang.demo.controller;
2
3 public class SinhVienController { no usages
4     |
5
6 }
```

Code mẫu class **SinhVienController.java**

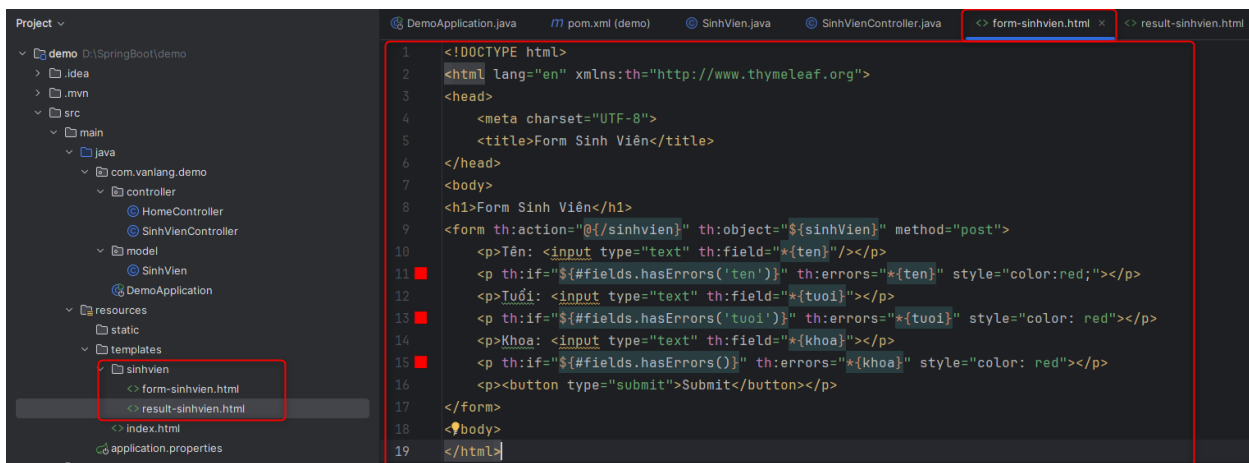


```
2
3 import com.vanlang.demo.model.SinhVien;
4 import jakarta.validation.Valid;
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.validation.BindingResult;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.PostMapping;
10
11 @Controller
12 public class SinhVienController {
13
14     @GetMapping("/sinhvien")
15     public String showForm(Model model) {
16         model.addAttribute("sinhVien", new SinhVien());
17         return "sinhvien/form-sinhvien";
18     }
19
20     @PostMapping("/sinhvien")
21     public String submitForm(@Valid SinhVien sinhVien, BindingResult bindingResult, Model model) {
22         if(bindingResult.hasErrors()){
23             return "form-sinhvien";
24         }
25         model.addAttribute("message", "Sinh viên đã được thêm thành công!");
26         return "sinhvien/result-sinhvien";
27     }
28 }
29
```

Bước 4: Tạo Templates Thymeleaf

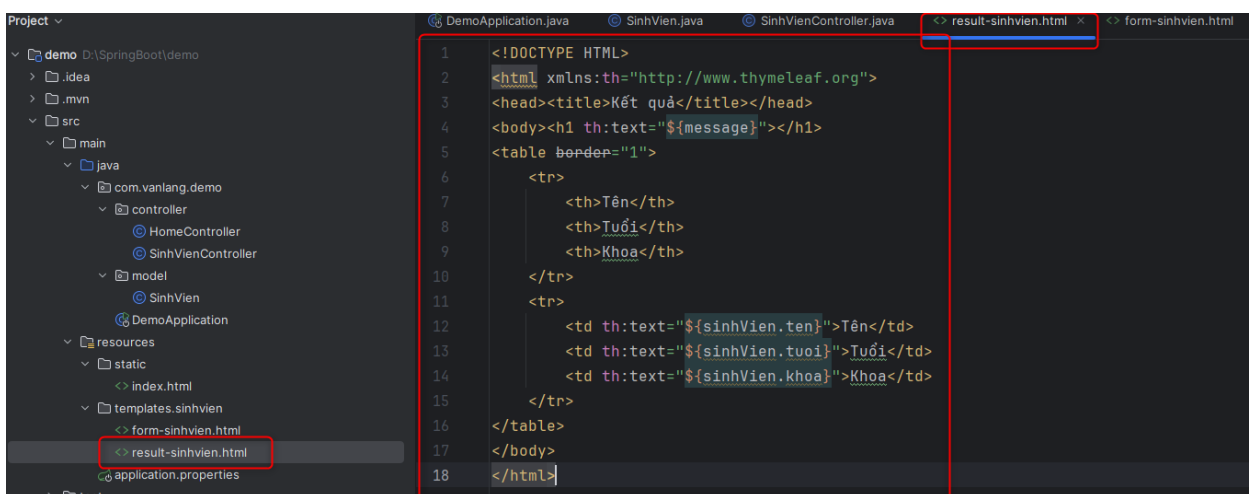
Trong package **templates** tạo thêm directory **sinhvien**. Trong directory **sinhvien** tạo thêm các trang hiển thị html: **form-sinhvien.html**, **result-sinhvien.html**,

Code mẫu form nhập liệu **form-sinhvien.html**:



```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <meta charset="UTF-8">
5 <title>Form Sinh Viên</title>
6 </head>
7 <body>
8 <h1>Form Sinh Viên</h1>
9 <form th:action="@{/sinhvien}" th:object="${sinhvien}" method="post">
10 <p>Tên: <input type="text" th:field="*{ten}" /></p>
11 <p th:if="${#fields.hasErrors('ten')}" th:errors="*{ten}" style="color:red;"></p>
12 <p>Tuổi: <input type="text" th:field="*{tuoi}" /></p>
13 <p th:if="${#fields.hasErrors('tuoi')}" th:errors="*{tuoi}" style="color: red;"></p>
14 <p>Khoa: <input type="text" th:field="*{khoa}" /></p>
15 <p th:if="${#fields.hasErrors()}" th:errors="*{khoa}" style="color: red;"></p>
16 <p><button type="submit">Submit</button></p>
17 </form>
18 </body>
19 </html>
```

Code mẫu trang hiển thị kết quả sinh viên ‘result-sinhvien.html’:

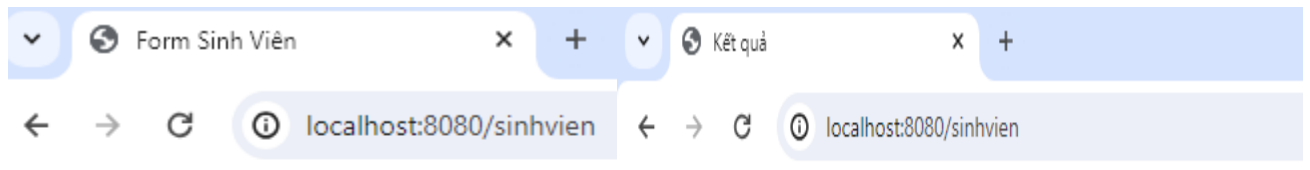


```
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head><title>Kết quả</title></head>
4 <body><h1 th:text="${message}"></h1>
5 <table border="1">
6 <tr>
7 <th>Tên</th>
8 <th>Tuổi</th>
9 <th>Khoa</th>
10 </tr>
11 <tr>
12 <td th:text="${sinhvien.ten}">Tên</td>
13 <td th:text="${sinhvien.tuoi}">Tuổi</td>
14 <td th:text="${sinhvien.khoa}">Khoa</td>
15 </tr>
16 </table>
17 </body>
18 </html>
```

KẾT QUẢ THAM KHẢO:

Trang nhập sinh viên

Trang hiển thị kết quả sinh viên



Form Sinh Viên

Tên:

Tuổi:

Khoa:

Sinh viên đã được thêm thành công!

Tên	Tuổi	Khoa
Nguyễn Minh Tân	20	CNTT

Yêu cầu bổ sung

Tạo trang Layout dùng chung, sử dụng Bootstrap cho Layout và sửa lại tất cả các trang Thêm sinh viên, Hiển thị kết quả sinh viên.

Hướng dẫn:

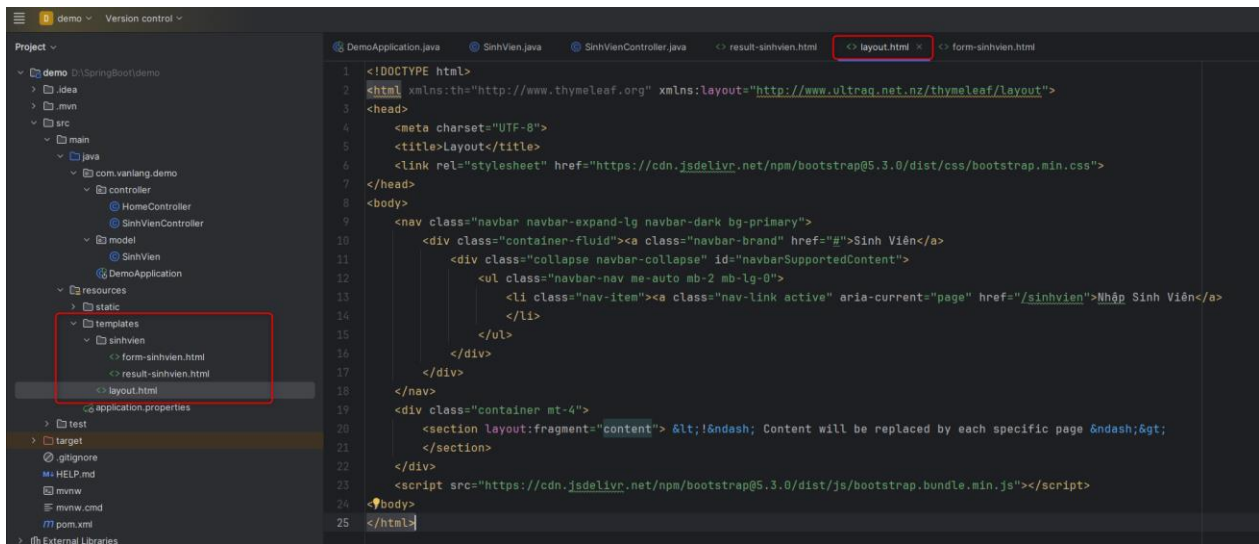
Đề tạo một trang layout dùng chung cho các trang khác trong ứng dụng Spring Boot và Thymeleaf, có thể sử dụng cơ chế layout của Thymeleaf. Điều này sẽ duy trì nhất quán và dễ dàng quản lý các thay đổi trên toàn bộ ứng dụng.

Bước 1: Tạo Layout Chính

Tạo một file mới tên là **layout.html** trong đường dẫn **src/main/resources/templates**

File này sẽ chứa cấu trúc **layout chính** cho các trang khác:

Code mẫu trang '**layout.html**':



Code tham khảo

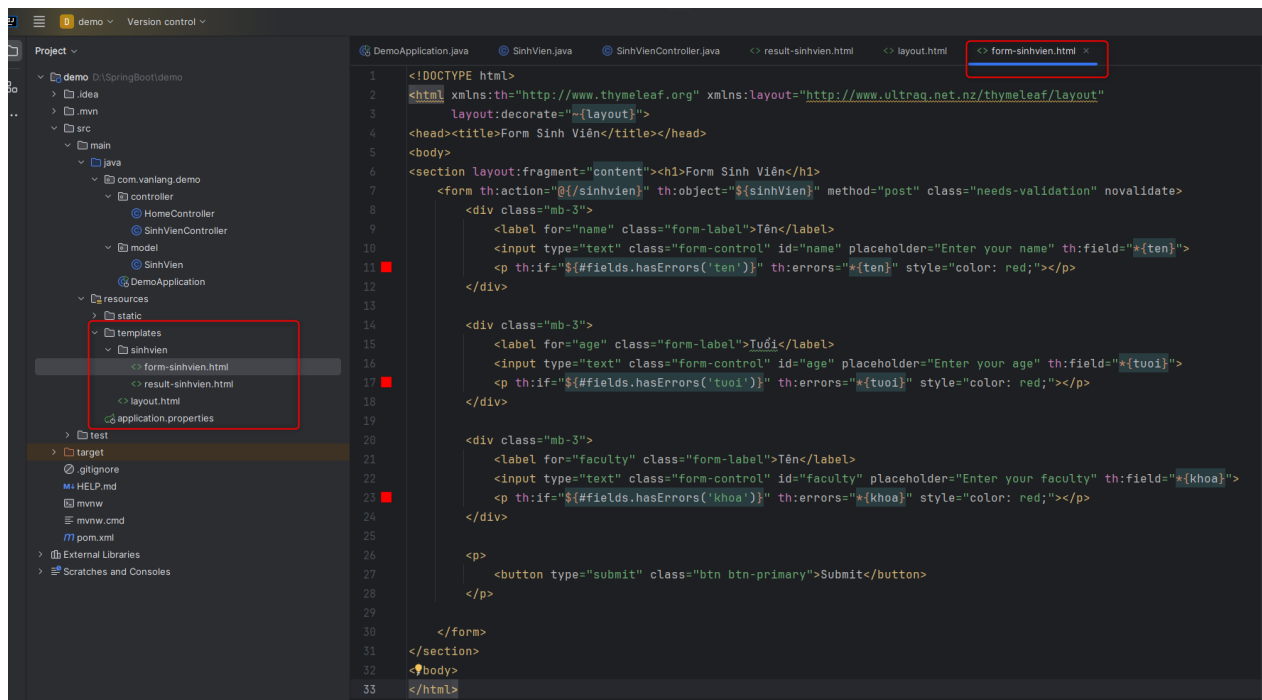
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">
<head>
  <meta charset="UTF-8">
  <title>Layout</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/boo
tstrap.min.css">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-
primary">
    <div class="container-fluid"><a class="navbar-brand"
href="#">Sinh Viên</a>
    <div class="collapse navbar-collapse"
id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item"><a class="nav-link
active" aria-current="page" href="/sinhvien">Nhập Sinh Viên</a>
      </li>
      </ul>
    </div>
  </div>
  <div class="container mt-4">
    <section layout:fragment="content"> &lt;!&dash;
Content will be replaced by each specific page &dash;&gt;
  </section>
  </div>
```



```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootst
trap.bundle.min.js"></script>
</body>
</html>
```

Bước 2: Sử Dụng Layout Trong Các Trang Khác

Form Sinh Viên ‘form-sinhvien.html’:



Code tham khảo

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
layout:decorate="~{layout}">
<head><title>Form Sinh Viên</title></head>
<body>
<section layout:fragment="content"><h1>Form Sinh Viên</h1>
<form th:action="@{/sinhvien}" th:object="${sinhVien}"
method="post" class="needs-validation" novalidate>
<div class="mb-3">
<label for="name" class="form-label">Tên</label>
<input type="text" class="form-control" id="name"
placeholder="Enter your name" th:field="*{ten}">
<p th:if="${#fields.hasErrors('ten')}" th:errors="*{ten}" style="color: red;"></p>
</div>
<div class="mb-3">
<label for="age" class="form-label">Tuổi</label>
<input type="text" class="form-control" id="age" placeholder="Enter your age" th:field="*{tuoi}">
<p th:if="${#fields.hasErrors('tuoi')}" th:errors="*{tuoi}" style="color: red;"></p>
</div>
<div class="mb-3">
<label for="faculty" class="form-label">Tên</label>
<input type="text" class="form-control" id="faculty" placeholder="Enter your faculty" th:field="*{khoa}">
<p th:if="${#fields.hasErrors('khoa')}" th:errors="*{khoa}" style="color: red;"></p>
</div>
<p>
<button type="submit" class="btn btn-primary">Submit</button>
</p>
</form>
</section>
</body>
</html>
```

```
th:errors="*{ten}" style="color: red;"></p>
</div>

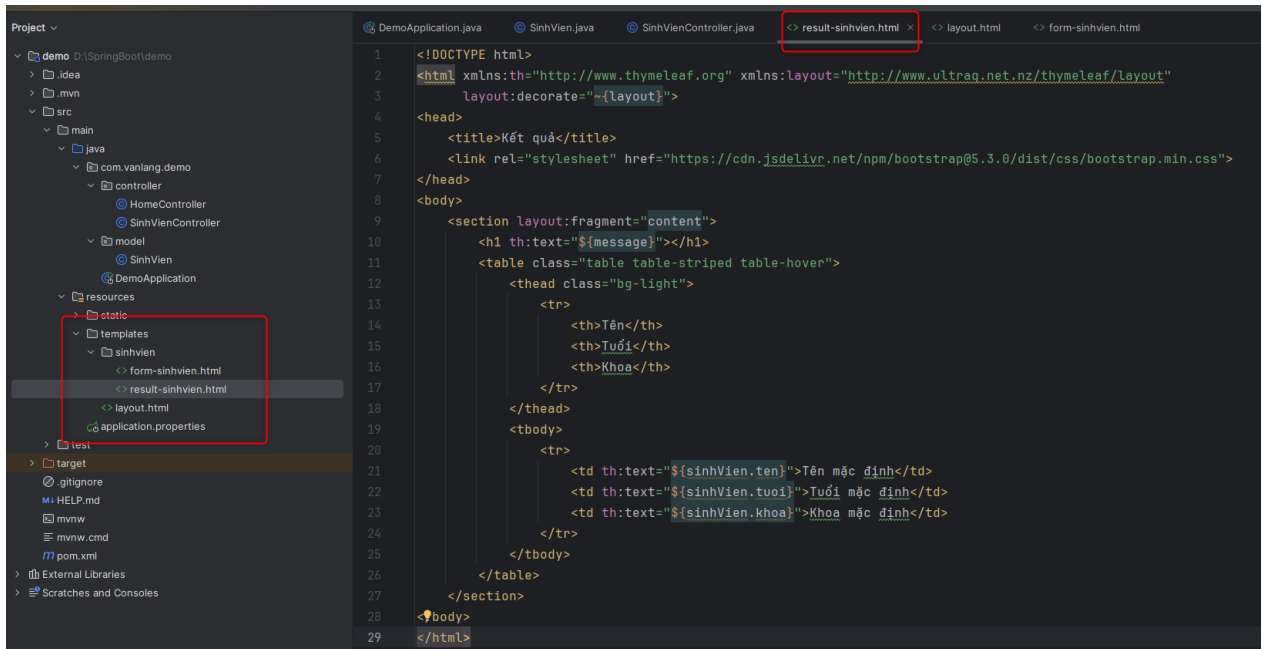
<div class="mb-3">
    <label for="age" class="form-label">Tuổi</label>
    <input type="text" class="form-control" id="age"
placeholder="Enter your age" th:field="*{tuoi}">
    <p th:if="${#fields.hasErrors('tuoi')}}"
th:errors="*{tuoi}" style="color: red;"></p>
</div>

<div class="mb-3">
    <label for="faculty" class="form-label">Tên</label>
    <input type="text" class="form-control"
id="faculty" placeholder="Enter your faculty"
th:field="*{khoa}">
    <p th:if="${#fields.hasErrors('khoa')}}"
th:errors="*{khoa}" style="color: red;"></p>
</div>

<p>
    <button type="submit" class="btn btn-
primary">Submit</button>
</p>

</form>
</section>
</body>
</html>
```

Trang Kết Quả ‘result-sinhvien.html’:



Code tham khảo

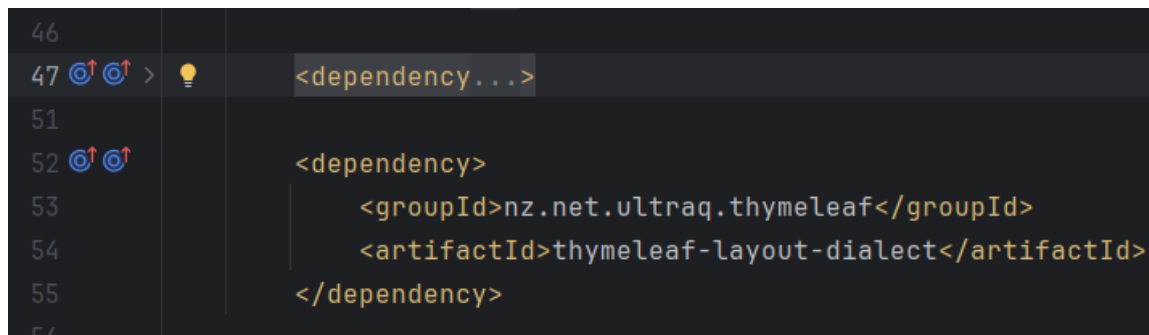
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
  layout:decorate="~{layout}">
<head>
  <title>Kết quả</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/boo
tstrap.min.css">
</head>
<body>
  <section layout:fragment="content">
    <h1 th:text="${message}"></h1>
    <table class="table table-striped table-hover">
      <thead class="bg-light">
        <tr>
          <th>Tên</th>
          <th>Tuổi</th>
          <th>Khoa</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td th:text="${sinhVien.ten}">Tên mặc
định</td>
          <td th:text="${sinhVien.tuoi}">Tuổi mặc
```

```
định</td>
                <td th:text="${sinhVien.khoa}">Khoa mặc
định</td>
            </tr>
        </tbody>
    </table>
</section>
</body>
</html>
```

Bước 3: Cài đặt dependency Thymeleaf Layout Dialect vào pom.xml

Lưu ý: Đảm bảo rằng đã cài đặt **thymeleaf-layout-dialect** trong dự án của bạn để sử dụng tính năng **layout** của **Thymeleaf**:

```
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>
```



The screenshot shows an IDE with a Maven pom.xml file open. Line 47 has a snippet "<dependency...>". Line 52 shows the start of a new dependency block: "<dependency>". Lines 53 and 54 show the groupId and artifactId being added: "<groupId>nz.net.ultraq.thymeleaf</groupId>" and "<artifactId>thymeleaf-layout-dialect</artifactId>". Line 55 shows the closing tag: "</dependency>".

Sau khi thêm **Dependency** ta tiến hành nhấp chuột phải → chọn **Maven** → chọn **Reload project** để dự án tải và cập nhật **Thymeleaf Layout Dialect** vào dự án.

Kết quả tham khảo:

- Trang form

Form Sinh Viên

Tên
Nguyễn Minh Tân

Tuổi
20

Khoa
CNTT

Submit

- Trang result:

Sinh viên đã được thêm thành công!

Tên	Tuổi	Khoa
Nguyễn Minh Tân	20	CNTT