**SPRING MVC SPRING BOOT**

Logo, company name

Description automatically generated

1. **Các bước tạo project và cấu hình :**

* Trang tham khảo

(https://laptrinhjavaweb.com/huong-dan-tao-project-spring-boot-co-ban-10#tao-project-spring-boot-01)

1. Mở Eclipse EE, sau đó File -> New -> other
2. Hộp thoại mở ra, làm theo như trong hình

Graphical user interface, text, application

Description automatically generated

1. Tiếp tuc làm theo như hình

Graphical user interface, text, application

Description automatically generated

4. Tiếp tuc làm theo như hình

1. Graphical user interface, application

   Description automatically generated
2. Anh chị chú ý Artifact Id anh chị để theo tên project anh chị đang làm
3. 5. Mở file POM.xml và thêm đoạn code như bên dưới
4. <parent>
5. <groupId>org.springframework.boot</groupId>
6. <artifactId>spring-boot-starter-parent</artifactId>
7. <version>1.5.22.RELEASE</version>
8. </parent>
9. <properties>
10. <java.version>1.8</java.version>
11. </properties>
12. <dependencies>
13. <dependency>
14. <groupId>org.springframework.boot</groupId>
15. <artifactId>spring-boot-starter-web-services</artifactId>
16. </dependency>
17. </dependencies>

6. Sau đó update project như hình bên dưới

Graphical user interface, application

Description automatically generated with medium confidence

Sau khi update anh chị làm theo các hình bên dưới nhé để build project

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Lệnh mvn clean install giúp anh chị xóa file JAR cũ và build file JAR mới để update code mới

7. Sau khi build project xong thì làm tiếp theo hình bên dưới

Graphical user interface, text, application

Description automatically generated

Trong folder src/main/java tạo 1 package có tên: com.laptrinhjavaweb

Trong package đó, tạo 1 class java Application.java và copy đoạn code bên dưới vào file Application.java

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

2. @SpringBootApplication
3. **public** **class** Application {
4. **public** **static** **void** main(String[] args) {
5. SpringApplication.run(Application.**class**, args);
6. }
7. }

8. Tiếp tục trong package com.laptrinhjavaweb tạo 1 package con là api. Sau đó trong package api tạo 1 file java NewAPI.java và copy đoạn code bên dưới vào file NewAPI.java

1. Graphical user interface, application

   Description automatically generated with medium confidence
2. **import** org.springframework.web.bind.annotation.GetMapping;
3. **import** org.springframework.web.bind.annotation.RestController;
5. @RestController
6. **public** **class** NewAPI {
7. @GetMapping("/test")
8. **public** String testAPI() {
9. **return** "success";
10. }
11. }
12. 9. Tiếp tục làm theo hình bên dưới

Graphical user interface, text, application

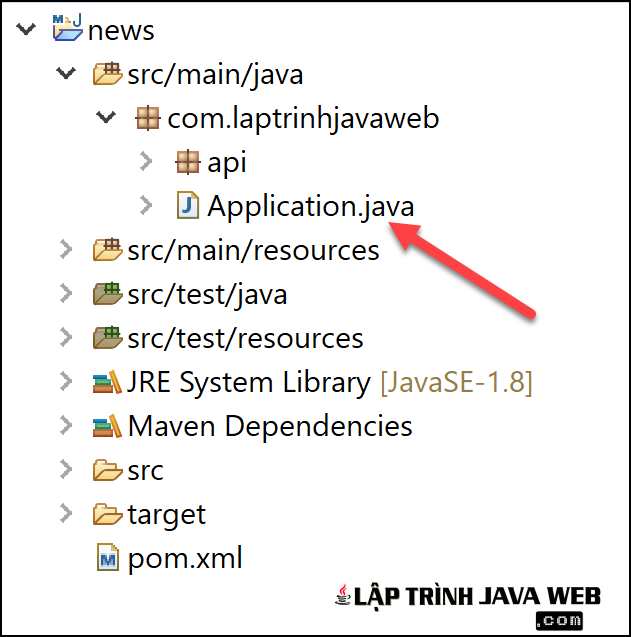
Description automatically generated

Trong folder src/main/resources tạo file application.properties và add đoạn code bên dưới vào file

**server.port**=**8081**

Mặc định port có sẵn của tomcat trong spring boot là 8080. Nhưng ta đổi thành port khác để tránh trùng với 1 số app nên chuyển từ 8080 -> 8081

**10. Sau khi hoàn thành các bưới kể trên, run project**



Kết quả thành công khi bạn mở trình duyệt và truy cập vào link

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Các bước thực hiện và giải thích các thành phần của project:**
2. Tạo các model tương ứng với các bảng ở CSDL dùng JPA để tạo

* Bảng New:

Graphical user interface, text, application

Description automatically generated

* Bảng category :

Graphical user interface, text, application, email

Description automatically generated

+ @Entity : định nghĩa cái class này sẽ là 1 bảng trong CSDL

+ @Table(name=””) chỉ định tên table trong CSDL

+ @Column tạo column

+ vì bảng Category có quan hệ 1-n với New nên ta dùng @OneToMany

* Phần này đơn giản chỉ như vậy , trọng tâm là demo SpringMVC SpringBoot

1. Tạo câu truy vấn sử dụng Spring Data Jpa trong Spring FW:

A picture containing text

Description automatically generated

* Để sử dụng được cú pháp của Spring Data Jpa ta cần tạo 1 Interface extend JpaRepository<class Entity, type Id>
* Cú pháp đặt tên theo yêu cầu chuẩn của jpa , đúng với tên field mà ta định nghĩa trong entity .
* Cái này làm nhiều sẽ có kinh nghiệm thôi , ở trên là 2 ví dụ : tìm title của New và mã thể loại.

1. Tầng Service xử lý Logic cho chức năng thêm và sửa:

Graphical user interface, text, application, email

Description automatically generated

* @Service : khi run ứng dụng thì SpringIOC(inversion of control) sẽ quét qua và giúp ta đánh dấu class đó là 1 Bean đưa vào container và duy trì nó . Khi muốn dùng thì gọi ra dùng không cần phải khởi tạo lại ( giống như từ khóa static)
* @Autowired : Sử dụng cơ chế DI( dependencies injection) trong Spring FW giúp ta vào trong IOC container lấy cái bean đó ra nhúng vào nơi ta cần sử dụng.
* Vì có thay đổi dữ liệu ở CSDL nên ta cần @Transaction giúp ta bảo toàn dữ liệu khi thực hiện thất bại .
* Chức năng Thêm Mới , Sửa đổi :
  + Đầu tiên ta tìm kiếm thể loại theo mã thể loại mà User đã chọn
  + Tiếp theo ta kiểm tra xem model có tồn tại Id hay kh nếu không thì ta sẽ thực hiện chức năng thêm mới . Bằng cách gọi hàm save() được build sẵn của Spring Data Jpa .(setCategory lưu lại thể loại )
  + Ngược lại nếu tồn tại Id thì ta sẽ thực hiện sửa đổi
  + Bằng cách tìm cái model ở trong CSDL thông qua Id gửi về , và ta set các thuộc tính ở model user gửi về và save().

1. **Luồng thực thi khi thêm mới hoặc sửa đổi:**

**Graphical user interface, text, application

Description automatically generated**

* Khi client gửi một request , thì nó sẽ tìm đến cái url phù hợp thì nhảy vào . Để nhận biết được thì ta cần phải sử dụng @Controller( nhận biết nó là cái bean controller trong spring để nhảy vào xử lý ) và @RequestMapping( thuộc tính **value** để định nghĩa request URL và thuộc tính **method** để định nghĩa HTTP request method.)
* Trong method ta có sử dụng @ModelAttribute để map 1-1 các thuộc tính bên ngoài view với model của ta, và @RequestParam dùng để lấy value trên URL và trong form với tên trùng với tên tham số mà ta khai báo.
* Sau khi biết được method nào xử lý thì controller sẽ gọi đến service để xử lý các logic thêm , sửa . Nếu thành công thì nó sẽ redirect đến trang danh sách.