TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỀ TÀI**

**TÌM HIỂU VỀ SPRING BOOT**

*Người hướng dẫn*: **THẦY** **NGUYỄN CHÍ THIỆN**

*Người thực hiện*: **LÊ NHỰT KHÁNH - 51503360**

**PHẠM QUỐC THẮNG - 51503226**

Lớp **: 15050301**

Khoá  **: 19**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỀ TÀI**

**TÌM HIỂU VỀ SPRING BOOT**

*Người hướng dẫn*: **THẦY NGUYỄN CHÍ THIỆN**

*Người thực hiện*: **LÊ NHỰT KHÁNH - 51503360**

**PHẠM QUỐC THẮNG - 51503226**

Lớp **: 15050301**

Khoá  **: 19**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018**

LỜI CẢM ƠN

Trước hết tôi xin tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Chí Thiện, thầy đã tận tình hướng dẫn, giúp đỡ tôi rất nhiều trong suốt quá trình thực hiện luận văn này. Sự hiểu biết sâu sắc về khoa học cũng như kinh nghiệm của thầy chính là tiền đề giúp tôi hoàn thành luận văn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm để tài của riêng chúng tôi và được sự hướng dẫn của Thầy Nguyễn Chí Thiện;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Lê Nhựt Khánh*

*Phạm Quốc Thắng*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Khởi tạo một project với Spring Boot: biết được cách tạo một ứng dụng Web với Spring Boot và sự dễ dàng hơn trong việc thiết lập và phát triển ứng dụng so với các Spring khác.

Sử dụng Spring với JPA: Dùng JPA để thao tác với cơ sở dữ liệu trên Spring

Cách sử dụng Katharsis JSON API: Katharsis định nghĩa các tài nguyên có thể được chia sẻ trên một giao diện RESTful và một repository để xử lý chúng.

MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc387692905)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc387692906)

[TÓM TẮT ii](#_Toc387692907)

[MỤC LỤC 2](#_Toc387692908)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc387692909)

[CHƯƠNG 1 – MỞ ĐẦU 2](#_Toc387692910)

[1.1 Tiểu mục cấp 1 2](#_Toc387692911)

[1.1.1 Tiểu mục cấp 2 2](#_Toc387692912)

[1.1.1.1 Tiểu mục cấp 3 2](#_Toc387692913)

[1.1.1.2 Tiểu mục cấp 3 tiếp theo. 2](#_Toc387692914)

[1.1.2 Tiểu mục cấp 2 tiếp theo 2](#_Toc387692915)

[1.2 Nội dung của chương này 2](#_Toc387692916)

[CHƯƠNG 2 – TỔNG QUAN 2](#_Toc387692917)

[1.1 Trình bày công thức toán học 2](#_Toc387692918)

[1.2 Trình bày một hình vẽ, sơ đồ 2](#_Toc387692919)

[CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM 2](#_Toc387692920)

[3.1 Chèn bảng: 2](#_Toc387692921)

[3.2 Viết tắt 2](#_Toc387692922)

[3.3 Trích dẫn 2](#_Toc387692923)

[3.3.1 Tài liệu tham khảo và cách trích dẫn 2](#_Toc387692924)

[3.3.2 Qui định của Khoa Công nghệ thông tin 2](#_Toc387692925)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Kiến trúc FTP 2](#_Toc387689394)

**DANH MỤC BẢNG**

[Bảng 3.1 Ví dụ cho chèn bảng 2](#_Toc387689363)

CHƯƠNG 1 – TỔNG QUAN VỀ SPRING – SPRING BOOT

Giới thiệu về Spring

Khái niệm

Spring Framework, hay ngắn hơn là Spring, là một cấu trúc dùng để xây dựng và phát triển ứng dụng mã nguồn mở dành cho ngôn ngữ lập trình Java và được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring không bắt buộc người ta phải tuân theo một mô hình lập trình (programming model) cụ thể nào, nó như là một hình thức chủ yếu thay thế cho mô hình Enterprise JavaBean. Theo thiết kế, framework này giải phóng lập trình viên dùng Java, cho phép họ nhiều quyền tự do hơn, và đồng thời cung cấp một giải pháp tiện lợi, đầy đủ dẫn chứng bằng tài liệu, dễ dàng sử dụng, phù hợp với những thực hành thông dụng trong công nghệ phần mềm.

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

Phát triển một ứng dụng Spring

Để phát triển một ứng dụng đơn giản như “Hello World” bằng Spring những điều cần làm là:

Tạo một project sử dụng **Maven** với các dependency cần thiết của **Spring MVC** và **Servlet API**.

Một tập tin **web.xml** để khai báo **DispatcherServlet** của **Spring MVC**.

Một tập tin cấu hình của **Spring MVC**.

Một class **Controller** trả về một trang “Hello World” khi có request đến.

Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.

Kiến trúc, các module của Spring Framework

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó. Dưới đây là kiến trúc tổng thể của Spring Framework.

Hình 1 Kiến trúc, các module của Spring Framework

Test

Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.

Spring Core Container

Bao gồm các module spring core, beans, context và expression languate (EL)

Spring core, bean cung cấp tính năng IOC và Dependency Injection.

Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.

Expression Language được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

AOP, Aspects and Instrumentation

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

Data Access / Integration

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

Web

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

Spring còn phát triển nhiều project con như:

Spring MVC, Spring Security, Spring Boot, Spring Batch, Spring Social, Spring IO, Spring Cloud, Spring Mobile, Spring for Android, Spring Session

Trong đó Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

Spring Boot

Spring boot là gì và tại sao phải sử dụng nó?

SpringBoot là một framework giúp xây dựng các ứng dụng dựa trên Spring một cách nhanh chóng và dễ dàng. Nó sinh ra với mục tiêu chính là giảm thiếu (hoặc hoàn toàn) các cấu hình lùm xùm trong Spring, giúp những lập trình viên có thể tập trung vào business logic hơn từ đó nâng cao chất lượng của sản phẩm. Tất cả những gì bạn cần là viết mã logic cho application hoặc rất rất ít các cấu hình nguyên thủy của Spring.

Những lợi ích của Spring Boot trong việc phát triển các ứng dụng Spring

- Tự động cấu hình (automatic configuration): Spring boot có thể tự động cấu hình những thứ cơ bản mà hầu hết các ứng dụng Spring đều phải có.

- Khởi tạo các phụ thuộc (starter dependencies): Bạn chỉ việc khai báo những loại tính năng bạn cần, nó sẽ đảm bảo các phụ thuộc cần thiết sẽ được thêm vào project.

- Giao diện command-line (The command-line interface): Đây là sự lựa chọn, Spring boot cho phép bạn chỉ cần viết code logic không cần xây dựng project như truyền thống.

- The Actuator: Nó cho phép bạn giám sát những gì đã xảy ra bên trong một ứng dụng đang chạy.

Những tiện ích đó sẽ giúp đơn giản hóa việc phát triển ứng dụng Spring theo cách riêng của mình.

Auto-configuration:

Trong bất cứ ứng dụng Spring nào, bạn sẽ phải cấu hình sử dụng Java code hoặc xml hoặc cả 2. Ví dụ bạn viết một ứng dụng Spring có kết nối tới csdl quan hệ thì có thể bạn sẽ cấu hình như sau:

@Bean

public JdbcTemplate jdbcTemplate(DataSource dataSource) {

return new JdbcTemplate(dataSource);

}

Đoạn code trên là cách đơn giản để tạo ra một bean JdbcTemplate và nó sẽ phải phụ thuộc vào DataSource. Vì vậy, DataSource sẽ được inject vào, điều đó có nghĩa là bạn phải tạo ra một DataSource bean nữa để inject vào đó. Giá sử bạn muốn sử dụng embedded H2 database vào thì DataSource bạn phải cấu hình như sau:

@Bean

public DataSource dataSource() {

return new EmbeddedDatabaseBuilder()

.setType(EmbeddedDatabaseType.H2)

.addScripts('schema.sql', 'data.sql')

.build();

}

Phương thức build() sẽ trả về một DataSource. Ui có vẻ dài dòng và rắc rối quá nhỉ? Nhưng đó chỉ là một phần nhỏ trong những gì mình phải cấu hình trong toàn ứng dụng.

Hơn nữa, mọi ứng dụng Spring khi sử dụng H2 thì để phải sử dụng 2 phương thức (tạo bean) trên. Điều đó có nghĩa là những cấu hình đó nó như là một cái khuôn cho các ứng dụng khác sử dụng H2. Nó phổ biến và mang tính chung chung như vậy, tại sao chúng ta phải bỏ công cấu hình trong mỗi lần tạo project?

Spring boot có thể tự động cấu hình những thứ tương tự vậy, nếu Spring boot phát hiện trong classpath của bạn có lib H2 thì nó sẽ tự cấu hình H2 vào ứng dụng của bạn. Nếu JdbcTemplate có trong classpath, nó sẽ cấu hình và có thể tự động inject vào bất cứ bean nào mà bạn viết. Vì vậy, chúng ta sẽ bỏ quả được một thứ khác rắc rối rồi. So good!

Starter dependencies:

Quản lý dependencies là một vấn đề khá rắc rối trong khi xây dựng project. Thư viện nào bạn sẽ cần thêm vào project? Group và Artifact của nó là gì? Version nào bạn cần? Sersion đó có phù hợp với các thư viện khác không? Có xung đột gì không?

Nó thật là khó chịu, tốn thời gian biết bao vì trong một project thì biết bao lib phụ thuộc, không lẽ chúng ta phải nhớ từng version của nó, rồi phải test thử có xung đột hay không, rồi lỡ có thì lại thay rồi lại test, cái vòng luẩn quẩn.

Spring boot hoàn toàn có thể giúp bạn việc này, ví dụ bạn muốn xây dựng một REST API sử dụng Spring trả về Json thì bạn sẽ phải cấu hình trong build.gradle (nếu dùng Gradle) hoặc POM.xml (nếu dùng Maven) các lib sau:

■ org.springframework:spring-core

■ org.springframework:spring-web

■ org.springframework:spring-webmvc

■ com.fasterxml.jackson. core:jackson-databind

■ org.hibernate:hibernate-validator

■ org.apache.tomcat.embed:tomcat-embed-core

■ org.apache.tomcat.embed:tomcat-embed-el

■ org.apache.tomcat.embed: tomcat-embed-logging-juli

Hoặc có thể bạn add bằng tay vào classpath :D. Có vẻ hơi mệt đây, đừng lo, Spring boot sẽ giúp bạn. Ví dụ bạn muốn xây dựng web application, bạn chỉ việc khai báo "web starter" org.springframework.boot:spring-boot-starter-web trong file cấu hình (build.gradle hoặc POM.xml).

Nếu bạn muốn dùng JPA thì khi báo "jpa starter" org.springframework.boot:spring-boot-starter-jpa. Spring sẽ tự động add các thư viện cần thiết vào classpath, các lib này đã được test cẩn thận nên bạn không phải lo về vấn đề conflict hay version gì nữa. Hehe... quá đã!

Command-Line Interface:

Ngoài khả năng tự động cấu hình và khởi xướng các thư viện phụ thuộc (starter dependencies) để các lập trình viên có thể focus vào việc viết code hơn, thì Spring Boot cũng cung cấp một tính năng rất thú vị khác đó là Spring Boot CLI (Command-Line Interface).

Spring Boot CLI làm cho ứng dụng của chúng ta giảm lượng code đáng kể khi loại bỏ được số lượng lớn các dòng import rắc rối. Vậy CLI làm thế nào để biết RequestMapping hay RestController nằm trong packages nào và những class nào được thêm vào classpath?

Câu trả lời ngắn gọn đó là Spring boot CLI đã phát hiện các kiểu dữ liệu đó đang được sử dụng và nó biết starter dependencies nào được thêm vào để nó làm việc. Mỗi lần các dependencies đó được thêm vào classpath thì Spring Boot sẽ cấu hình những thứ cần thiết và chắc chắn rằng DispatcherServlet và Spring MVC đã được kích hoạt và các controller có thể phản hồi các HTTP request.

Spring Boot CLI là một sự lựa chọn nhỏ trong những gì mà Spring Boot có thể làm được. Mặc dù nó mang lại sức mạnh rất lớn và đơn giản hóa việc phát triển ứng dụng Spring, nhưng nó cũng được giới thiệu là một phong cách lập trình không theo nguyên tắc (unconvention development model). Nếu bạn không ưng phong cách này thì đó không phải là vấn đề, bạn có thể không sử dụng nó vì đơn giản, đây là một sự lựa chọn.

The Actuator:

Đây là một khả năng hữu ích trong những khả năng đáng chú ý của Spring Boot. Các thành phần trên nhằm đơn giản hóa việc phát triển ứng dụng Spring nhưng Actuator thì cung cấp khả năng kiểm tra những gì đã xảy ra bên trong một ứng dụng đang chạy, bao gồm những chi tiết như:

■ Những bean nào đã được cấu hình trong Spring application context

■ Những gì đã được Spring tự động cấu hình

■ Biến môi trường, system properties, configuration properties và command-line arguments nào có hiệu lực trong ứng dụng của bạn

■ Trạng thái hiện tại của threads và hỗ trợ ứng dụng của bạn

■ Chỉ ra HTTP request gần đây được thực hiện bởi ứng dụng

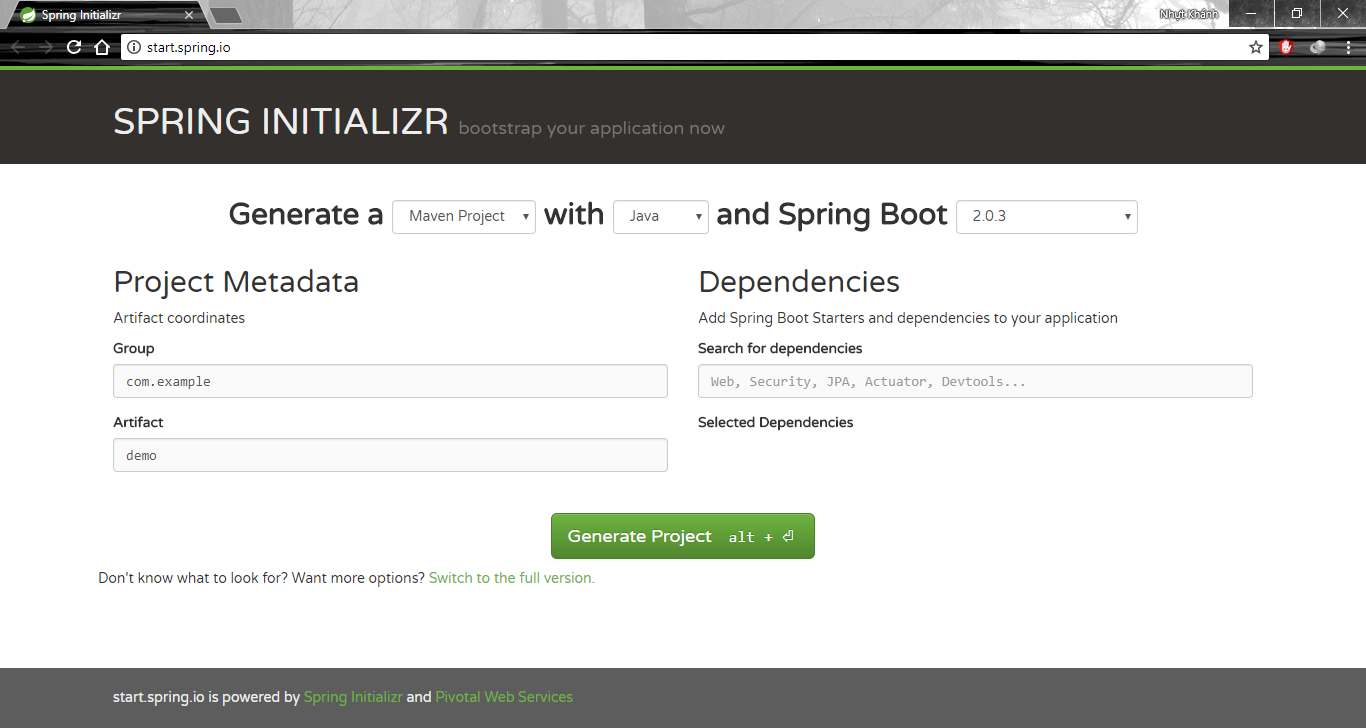
■ Các số liệu liên quan đến việc sử dụng bộ nhớ, bộ dọn rác, web requests và nguồn dữ liệu sử dụng

Actuator trình bày thông tin này bằng hai cách là thông qua web endpoints hoặc thông qua shell interface.

Phát triển với một ứng dụng Spring Boot

Sử dụng Spring Initializr Web

Bước 1: Truy cập vào trang web của **Spring Initializr Web: start.spring.io**

Hình 2 **Spring Initializr Web**

Chọn: **Maven Project**

Ngôn ngữ: **Java**

Phiên bản Spring Boot: **2.0.3**

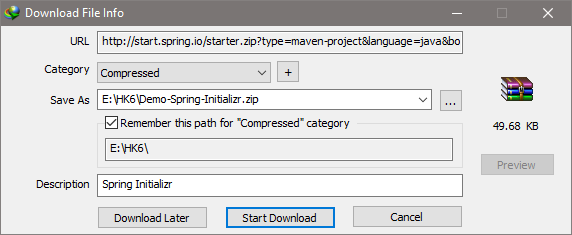
Group: **kientapcongnghiep**

Artifact: **Demo-Spring-Initializr**

Search for dependencies: **Web**

Chọn **Generate Project**

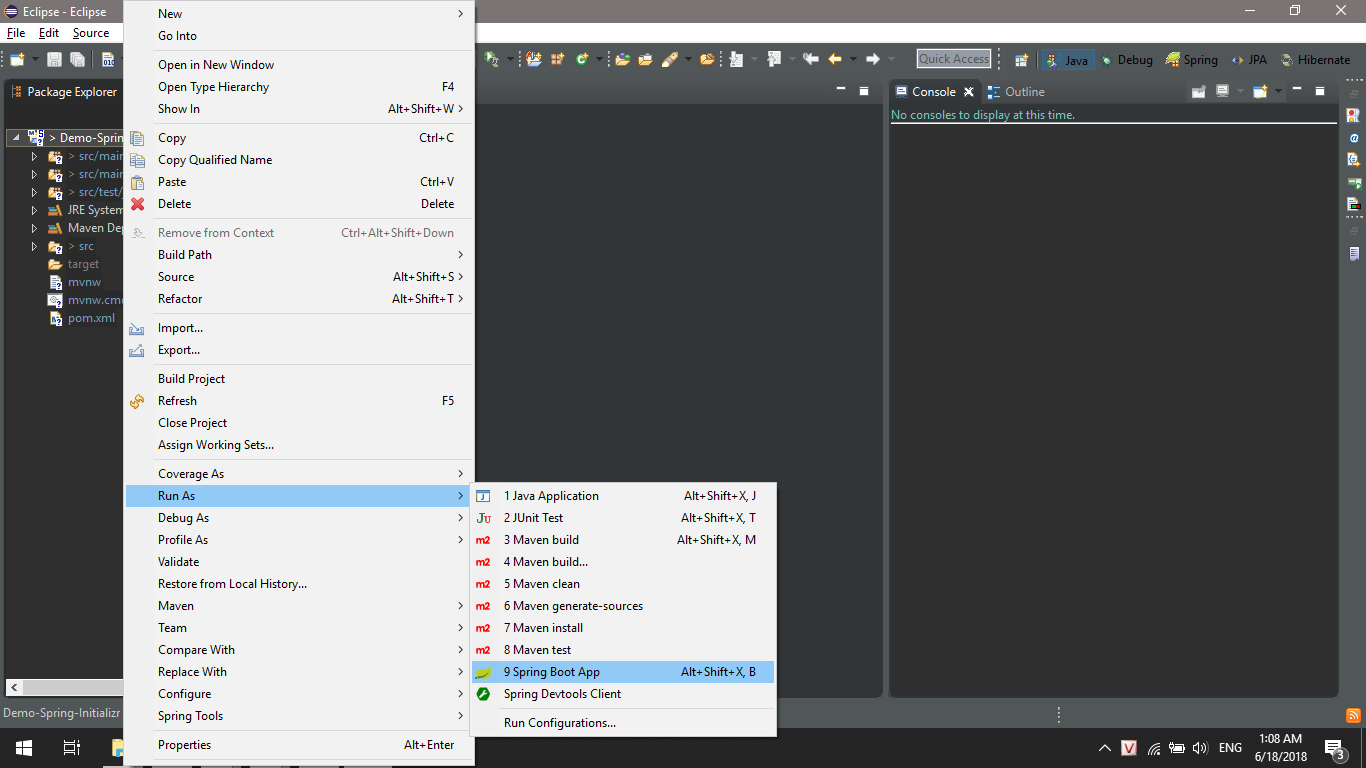
Một file vừa được tạo và tải xuống



Hình 3 Một project được tự động tải xuống

Giải nén và add project vừa được tạo vào Eclipse

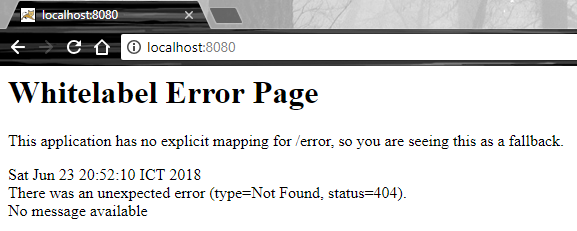
Click chuột phải vào Project vừa add chọn **Run as** chọn **Spring Boot Application**



Hình 4 Khởi chạy project

Như vậy một Spring Boot Project đã được tạo.

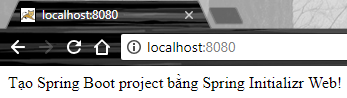
Mở trình duyệt và truy cập vào địa chỉ <http://localhost:8080>



Chúng ta đang nhìn thấy một lỗi là **404 Not Found**, nguyên nhân là vì trong project mà chúng ta vừa tạo không có một **Controller** nào xử lý cho request đến trang chủ.

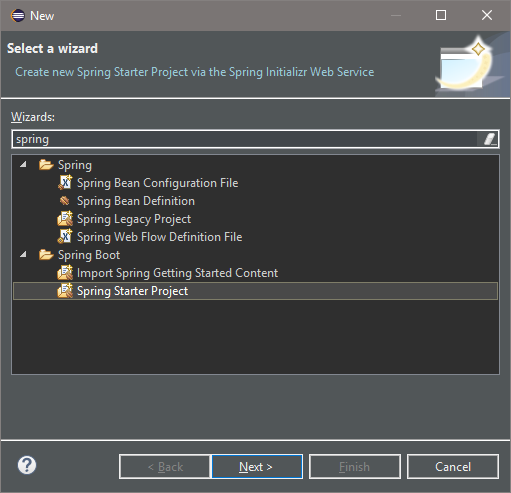
Lúc này project chưa có controller nên ta phải thêm vào

Sau đó chạy lại ứng dụng và truy cập vào địa chỉ <http://localhost:8080>



**Sử dụng công cụ Spring Tool Suite**

Tạo một Spring Starter Project



Name**: Demo-Spring-Tool-Suite**

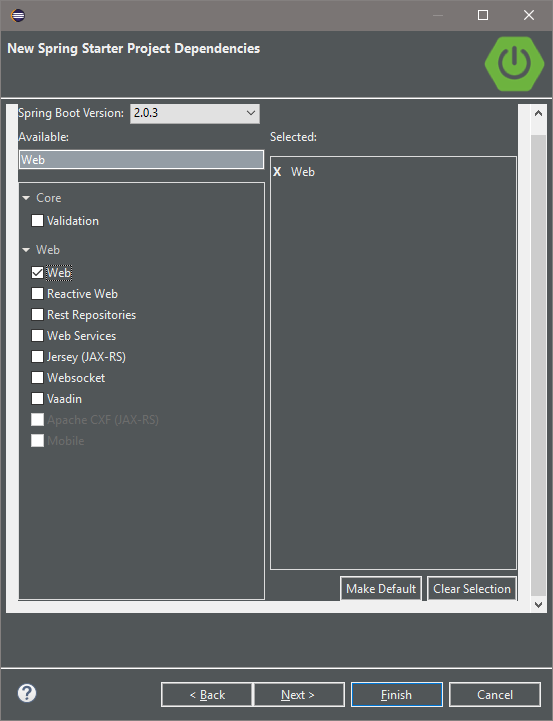
Location: **Chọn nơi lưu Project**

Type: **Maven**

Language: **Java**

Group: **kientapcongnghiep**

Artifact: **Demo-Spring-Tool-Suite**



Chọn Next chọn Spring Boot Version: **2.0.3**

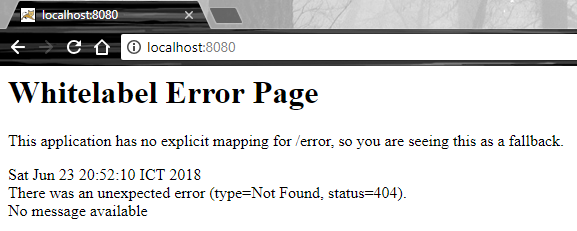
Tick vào **“Web”**

Chọn **Finish**

Như vậy một Spring Boot Project đã được tạo.

Click chuột phải vào Project vừa add chọn **Run as** chọn **Spring Boot Application**

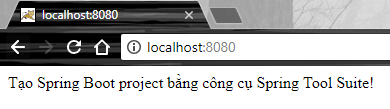
Mở trình duyệt và truy cập vào địa chỉ <http://localhost:8080>



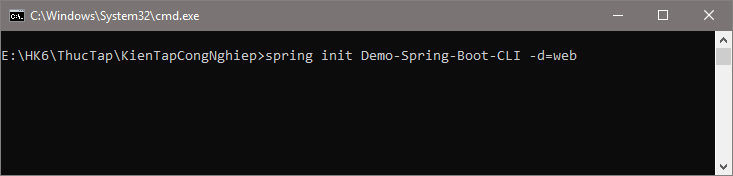
Chúng ta đang nhìn thấy một lỗi là **404 Not Found**, nguyên nhân là vì trong project mà chúng ta vừa tạo không có một **Controller** nào xử lý cho request đến trang chủ.

Lúc này project chưa có controller nên ta phải thêm vào

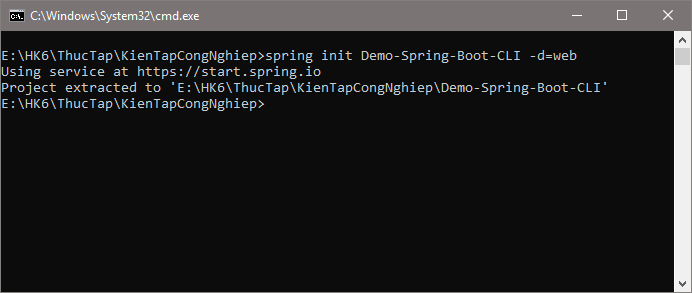
Sau đó chạy lại ứng dụng và truy cập vào địa chỉ <http://localhost:8080>



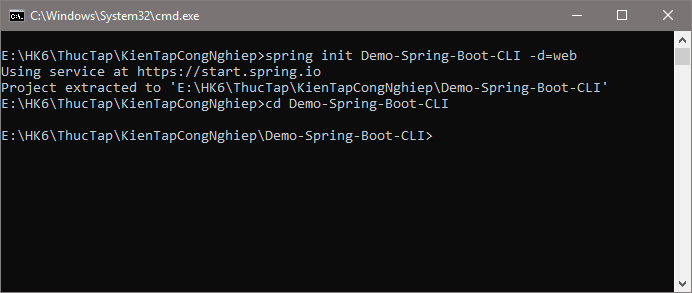
Dùng lệnh spring init Demo-Spring-Boot-CLI –d=web để tạo một Spring Boot project



Hình 5 Tạo project

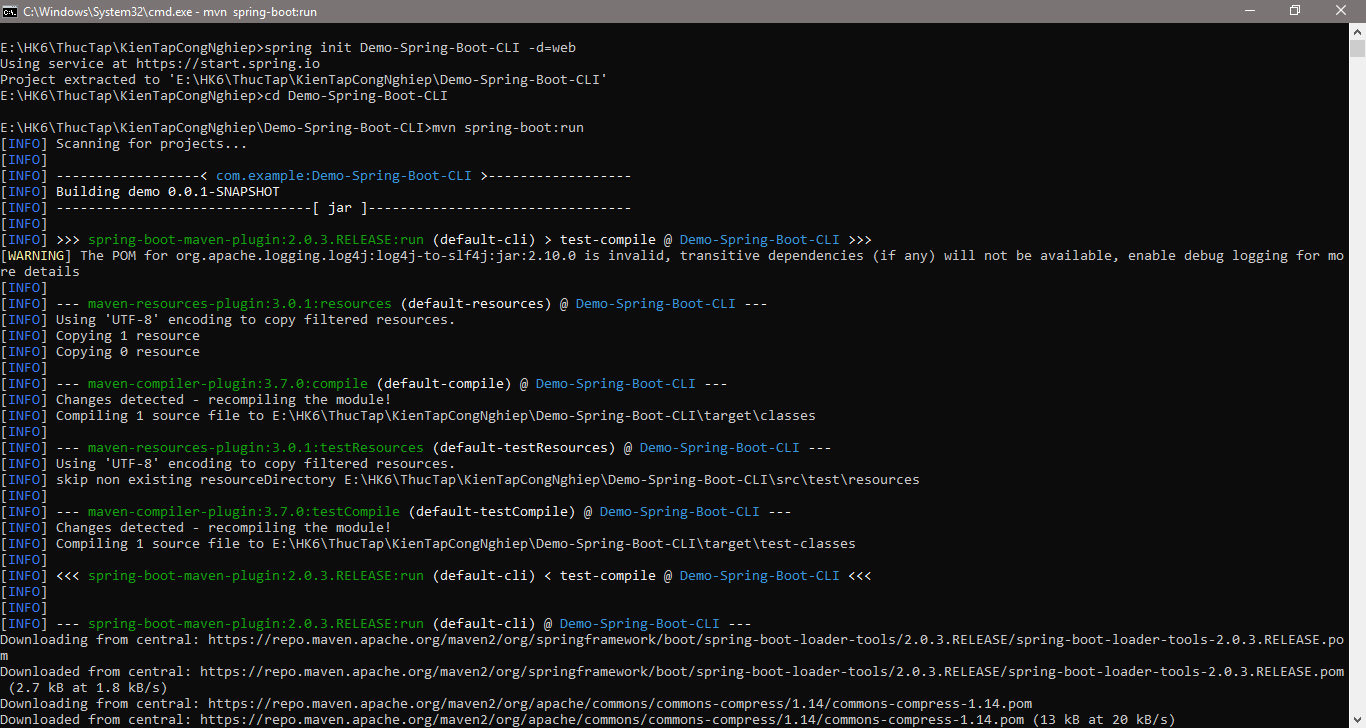


Hình 6 Project được tạo ra

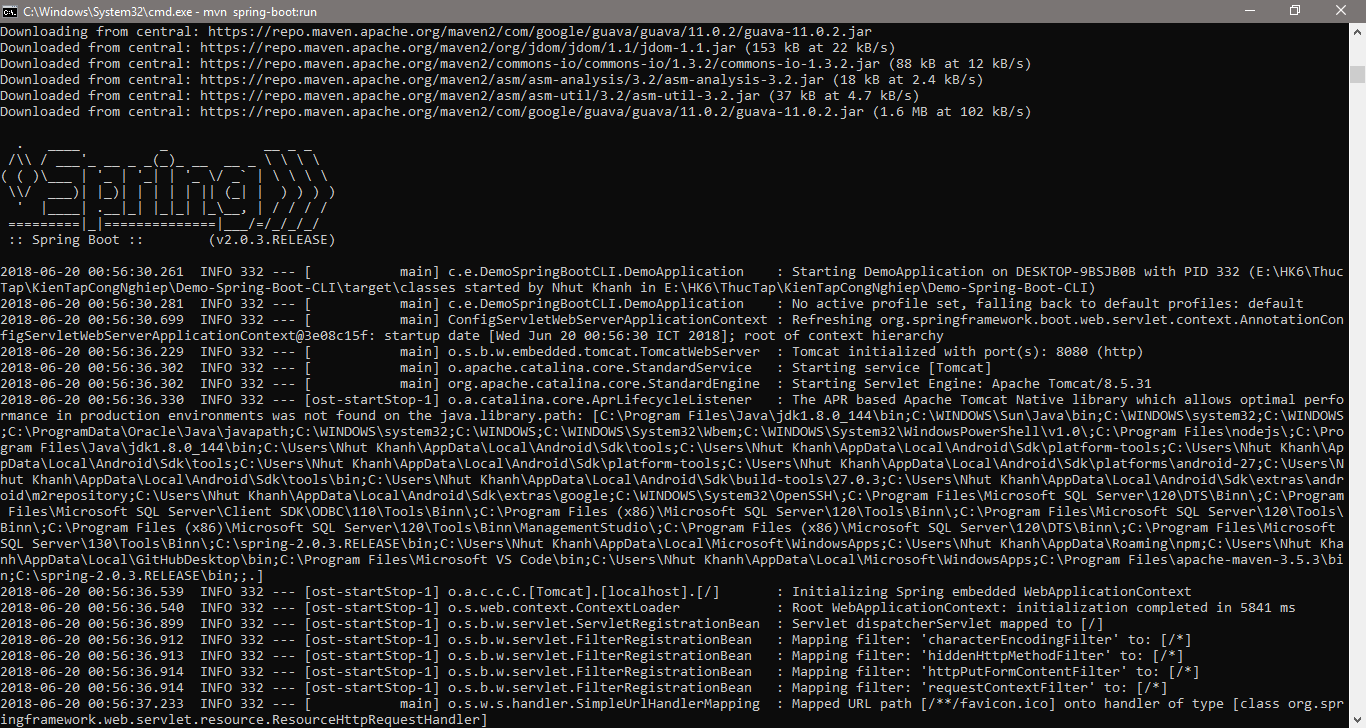


Hình 7 Chuyển đến thư mục làm việc của project

Để chạy ứng dụng ta dùng lệnh: mvn spring-boot:run

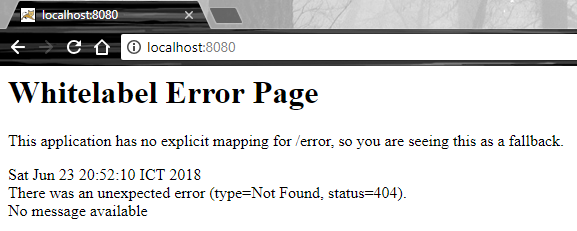


Hình 8 Chạy ứng dụng



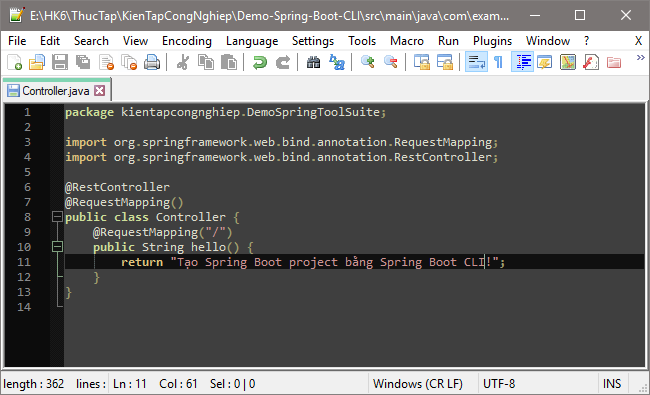
Hình 9 Chạy ứng dụng

Mở trình duyệt và truy cập vào địa chỉ <http://localhost:8080>



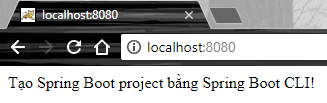
Chúng ta đang nhìn thấy một lỗi là **404 Not Found**, nguyên nhân là vì trong project mà chúng ta vừa tạo không có một **Controller** nào xử lý cho request đến trang chủ.

Lúc này project chưa có controller nên ta phải thêm vào



Hình 10 Controller

Sau đó chạy lại ứng dụng và truy cập vào địa chỉ <http://localhost:8080>



Ứng dụng minh họa nâng cao

Tổng quan về bài toán

Khi sinh viên hoặc nhân viên mới vào, họ sẽ được training theo 1 chủ đề nào đó (subject, for example: java). Trong subject thì có nhiều phân vùng (area, for example: jpa, java core, spring core, …) cần được training. Và sau mỗi phần, cần được làm bài kiểm tra, tức là làm bài kiểm tra dựa trên ngân hàng câu hỏi (questions) trong area nào đó.

Các chức năng chính của hệ thống như sau:

         Có thể thêm/xóa/sửa subject/area

         Có thể import questions

         Có thể thiết kế các bài kiểm tra

         Có thể giao bài cho user làm bài kiểm tra

         User có thể làm bài kiểm tra và biết được kết quả đạt/không đạt.

Các chức năng mở rộng của hệ thống như sau:

         Có thể thêm/xóa/sửa delivery group/ delivery center/ project

         Có thể tạo request resource từ các project

         Có thể import danh sách sinh viên từ các trường đại học gởi về

         Có thể tạo/xóa/sửa user cho các sinh viên/ nhân viên mới

         ….

Các bước xây dựng

training-lab-api project

Mở cmd ở thư mục cần lưu project

Gõ dòng lệnh sau: mvn archetype:generate -DgroupId=com.tma -DartifactId=training-lab-api -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

Trong đó:

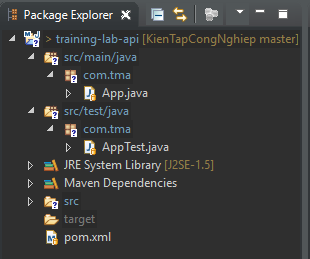
**archetype:generate** – tạo một project mới dựa trên một Archetype (khuôn mẫu) có sẵn

**DgroupId** – thường đặt theo tên của tổ chức hoặc nhóm tạo ra dự án: com.tma

**DartifactId** – thường lấy theo tên viết tắt của dự án, cũng là tên của project mà ta tạo: training-lab-api

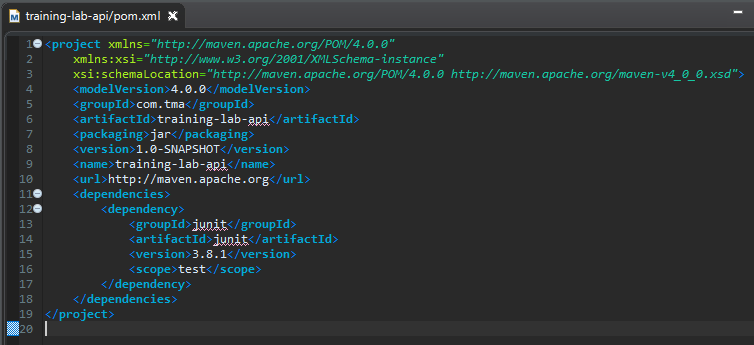
**DarchetypeArtifactId** – là loại dự án sẽ được tạo: Maven Quickstart project “để tạo cái maven của bạn với cấu trúc thư mục đơn giản và dành cho người mới học maven(đương nhiên là chỉ có đủ chức năng cho starter)”

Import project vừa tạo vào Eclipse ta được cấu trúc thư mục như sau:



Hình 11 Cấu trúc thư mục project

File pom.xml là nơi khai báo tất cả những gì liên quan đến dự án được cấu hình qua maven, như khai báo các dependency, version của dự án, tên dự án, repossitory, …



Hình 12 File pom.xml của project

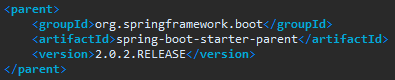
1.0-SNAPSHOT là version của project

Dependencies là những thư viện của dự án và được lưu ở thư mục Maven Dependencies ở cấu trúc thư mục hình 11.

Áp dụng Spring Boot vào training-lab-api project

Thêm parent vào file pom.xml

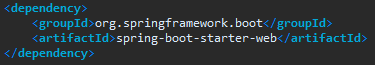
**spring-boot-starter-parent** là một project sẵn có trong **Spring Boot**. Các thư viện dependency cơ bản đã được khai báo trong **spring-boot-starter-parent**, project của bạn chỉ cần thừa kế nó. Chỉ cần khai báo **<parent>** trong file **pom.xml** của project.



Hình 13 Thêm parent vào file pom.xml

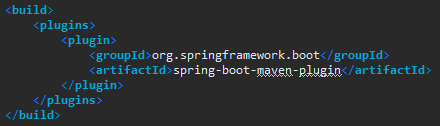
Do đang phát triển một json webservice nên chúng ta cần thêm dependency **spring-boot-starter-web vào file pom.xml**

Các **"Starter"** khác chỉ đơn giản là cung cấp phụ thuộc mà bạn có khả năng cần thiết khi phát triển một loại hình cụ thể của ứng dụng. Chẳng hạn khi bạn phát triển một ứng dụng web, bạn cần một dependency **spring-boot-starter-web**.



Hình 14 Thêm dependency **spring-boot-starter-web**

**Spring-boot-maven-plugin** là plugin cung cấp các thư viện cần thiết giúp project của bạn có thể chạy trực tiếp mà không cần triển khai trên một **Web Server**. Nó giúp tạo ra một file **jar** có thể thực thi (Executable)



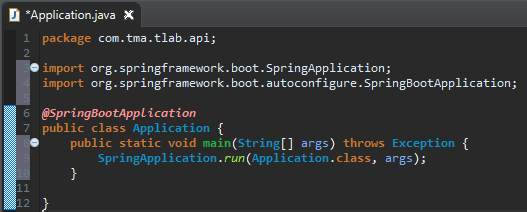
Hình 15 Thêm dependency spring-boot-maven-plugin

Xóa 2 class App và AppTest trong package com.tma

Tạo package tlab.api trong com.tma

Tạo class Application trong package vừa tạo

Chỉnh sửa class Application để trở thành trình khởi chạy với Spring Boot. Thêm @SpringBootApplication trước class Application và phương thức main().



Hình 16 Application.java

Để thay đổi port của ứng dụng Spring ta tạo file application.properties và sau đó thêm server.port = 9090 vào tệp này để thay đổi từ port 8080 sang 9090



Hình 17 application.properties

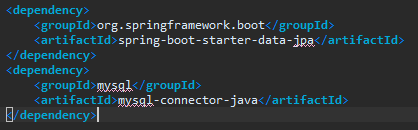
Sử dụng JPA để truy cập dữ liệu

Thêm một số thư viện cần thiết cho



Hình 18 Thêm thư viện cần thiết

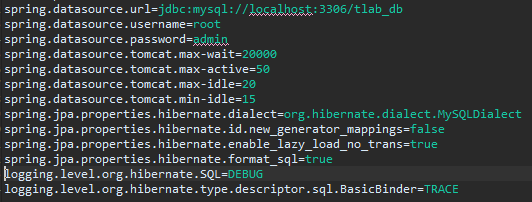
Thêm dependency spring-boot-starter-data-jpa để lưu trữ và truy xuất dữ liệu trong cơ sở dữ liệu quan hệ. Thêm dependency mysql-connector-java để kết nối với MySQL



Hình 19 Thư viện cho JPA và MySQL

Tạo một cơ sở dữ liệu là tlab\_db với các bảng như sau

Để **Spring** có thể kết nối vào Database thì cần phải cấu hình các thôngphairveef database và jpa trong tập tin **application.properties**



Hình 20 Cấu hình Database và JPA

Tạo package model để tạo các class tương ứng là một class đại diện cho một bảng trong cơ sở dữ liệu. Các trường trong lớp này sẽ tương ứng với các cột trong bảng.

Chúng ta sẽ tạo lớp **Subject** để đại diện cho bảng **subject** trong database. Các **JPA Annotation** sẽ được sử dụng để chú thích trên các trường để mô tả cách ánh xạ giữa các trường và các cột của bảng. Các ánh xạ này là 1-1, mỗi trường tương ứng với 1 cột trong bảng.

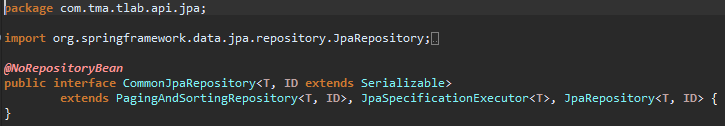


Hình 21 Subject.java

Các bảng còn lại tương tự

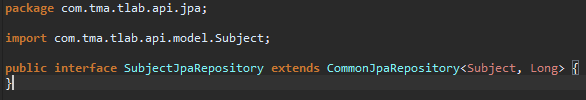
Tạo package jpa để ………….

Tạo interface CommonJpaRepository trong package com.tma.tlab.api.jpa

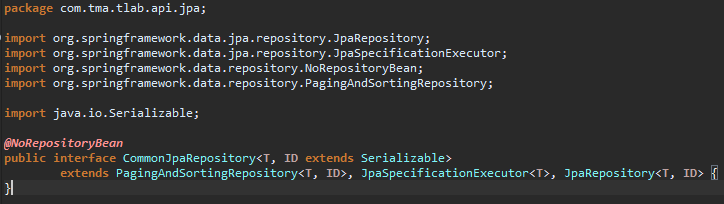


Hình 22 CommonJpaRepository.java

Tạo interface SubjectJpaRepository trong package com.tma.tlab.api.jpa



Hình 23 SubjectJpaRepository.java



Hình 24 CommonJpaRepository.java

Sử dụng Katharsis JSON API

Để sử dụng Katharsis JSON API cần phải thêm các dependency

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Quách Ngọc Ân (1992), “Nhìn lại hai năm phát triển lúa lai”, *Di tuyền học ứng dụng*, 98(1), tr. 10-16.
2. Bộ nông nghiệp & PTNT (1996), *Báo cáo tổng kết 5 năm (1992-1996) phát triển lúa lai,* Hà Nội.
3. Nguyễn Hữu Đống, Đào Thanh Bằng, Lâm Quang Dụ, Phan Đức Trực (1997), *Đột biến –* *Cơ sở lý luận và ứng dụng,* Nhà xuất bản nông nghiệp, Viện khoa học kỹ thuật nông nghiệp Việt Nam, Hà Nội.
4. Nguyễn Thị Gấm (1996), *Phát hiện và đánh giá một số dòng bất dục đực cảm ứng nhiệt* *độ,* Luận văn thạc sĩ khoa học nông nghiệp, Viện khoa học kỹ thuật nông nghiệp Việt Nam, Hà Nội.

……….

1. Võ Thị Kim Huệ (2000), *Nghiên cứu chẩn đoán và điều trị bệnh…,* Luận án Tiến sĩ y khoa, Trường đại học y Hà Nội, Hà Nội.

**Tiếng Anh**

1. Anderson J.E. (1985), The Relative Inefficiency of Quota, The Cheese Case, *American* *Economic Review*, 75(1), pp. 178-90.
2. Borkakati R. P.,Virmani S. S. (1997), Genetics of thermosensitive genic male sterility in Rice, *Euphytica* 88, pp. 1-7.
3. Boulding K.E. (1955), *Economics Analysis*, Hamish Hamilton, London.
4. Burton G. W. (1988), “Cytoplasmic male-sterility in pearl millet (penni-setum glaucum L.)”, *Agronomic Journal* 50, pp. 230-231.
5. Central Statistical Oraganisation (1995), *Statistical Year Book*, Beijing.
6. FAO (1971), *Agricultural Commodity Projections (1970-1980)*, Vol. II. Rome.
7. Institute of Economics (1988), *Analysis of Expenditure Pattern of Urban Households in* *Vietnam,* Departement pf Economics, Economic Research Report, Hanoi.

**PHỤ LỤC**

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn