

Bài 32

Spring Boot

Module: WBD-JAVA



Mục tiêu

- Giải thích được mô hình hoạt động của Spring Boot
- Cài đặt và sử dụng được Spring Boot để phát triển ứng dụng
- Cài đặt và sử dụng được Spring Boot CLI

Overview

Spring làm cho việc phát triển một ứng dụng Java trở lên dễ dàng...

Nhưng Spring Boot còn có thể làm nó dễ dàng hơn!

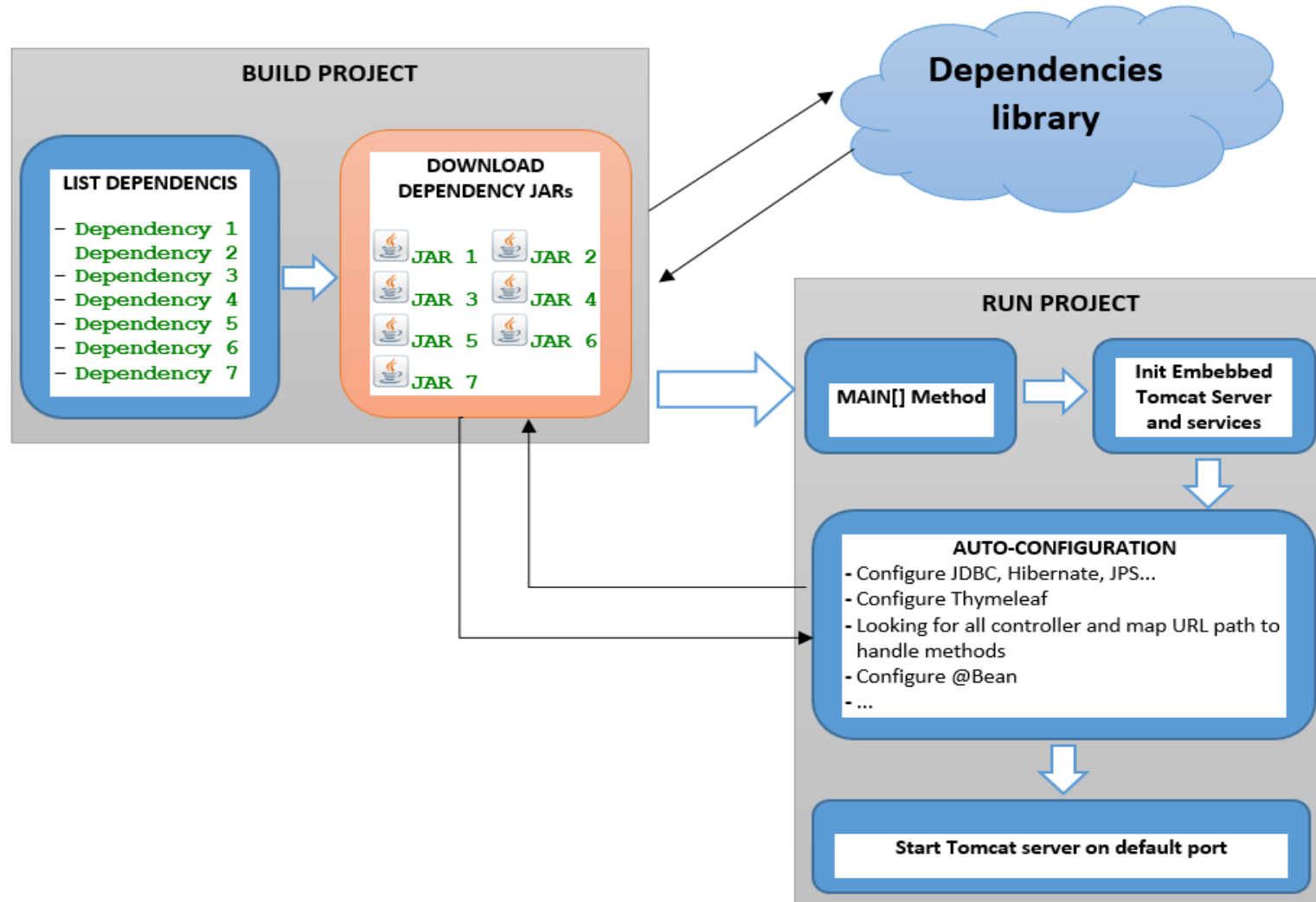
- Craig Walls -

Overview



- Spring Boot là giải pháp cho việc cấu hình một project độc lập của Spring.
- Spring Boot giúp cho việc bắt đầu một dự án Spring trở lên dễ dàng và nhanh chóng.
- Spring Boot được thể hiện thông qua một loạt các quy tắc nhằm mục đích.
 - Cung cấp ra các ứng dụng đã được cấu hình và đóng gói đầy đủ có thể chạy độc lập.
 - Giảm thiểu tối đa sự phức tạp của việc cấu hình project cũng như quản lý các thư viện được cung cấp từ bên thứ 3 ngoài Spring.

Các thức hoạt động



Cách thức hoạt động

- Cũng như các dự án Spring, Spring Boot sử dụng Gradle, Maven... để khai báo, quản lý và build các thư viện.
- Các dự án Spring Boot bắt đầu khởi chạy từ hàm main[] nằm trong một class được xác định thông qua annotation @SpringBootApplication

```
@SpringBootApplication
public class Application {
    |
    public static void main(String[] args) { SpringApplication.run(Application.class, args); }
}
```



Cách thức hoạt động

- Tomcat server được cấu hình nhúng sẵn trong một ứng dụng Spring Boot.
- Từ các thư viện dependencies đã khai báo, Spring Boot tự động cấu hình các thành phần cần thiết cho ứng dụng.

Đặc điểm(1)



- Spring Boot đưa ra 4 đặc điểm để phục vụ tối đa cho mục đích làm đơn và dễ dàng hơn trong việc phát triển dự án Spring:
 - Spring Boot starters
 - Auto-configuration
 - Command-line interface (CLI)
 - Actuator



Spring Boot starters

- Cung cấp một single dependency (spring-boot-starter) chứa những dependencies có chung đặc điểm.

Spring Boot starters



- Ví dụ:
 - Bắt đầu một ứng dụng RESTful Spring MVC, thao tác với dữ liệu từ database, kiểu dữ liệu là JSON và sử dụng Thymeleaf cho views, so sánh danh sách dependency giữa Spring MVC và Spring Boot.

Spring MVC	Spring Boot
<pre>dependencies { Compile("org.springframework:spring-web:4.0.6.RELEASE") Compile("org.springframework:spring-webmvc:4.0.6.RELEASE") Compile("com.fasterxml.jackson.core:jackson-databind:2.2.2") Compile("org.springframework:spring-jdbc:4.0.6.RELEASE") Compile("org.springframework:spring-tx:4.0.6.RELEASE") Compile("org.thymeleaf:thymeleaf-spring4:2.1.2.RELEASE") }</pre>	<pre>dependencies { Compile("org.springframework.boot:spring-boot-starter-web:1.1.4.RELEASE") Compile("org.springframework.boot:spring-boot-starter-jdbc:1.1.4.RELEASE") Compile("org.springframework.boot:spring-boot-starter-thymeleaf") }</pre>



Spring Boot starters

- Ưu điểm:
 - Rút ngắn danh sách dependency cần khai báo cho 1 dự án Spring.
 - Giảm bớt công việc phải làm trong quá trình khai báo và quản lý các dependencies.
 - Đảm bảo tất cả dependencies hoạt động ổn định cùng nhau.
 - Tạo sự rõ ràng, dễ đoán được tác dụng từ việc đặt tên các dependencies.

Auto-configuration

- Đặc điểm auto-configuration của Spring Boot sẽ tự động cấu hình dự án dựa trên các dependencies đã được thêm vào Gradle/Maven.
- Khởi động auto-configuration thông qua annotation `@EnableAutoconfiguration` hoặc `@SpringBootApplication`.
- Ví dụ:
 - Để sử dụng Thymeleaf cho việc xử lý các view trong một dự án:
 - Với SpringMVC cần cấu hình tối thiểu 3 đối tượng *ThymeleafView-Resolver*, *SpringTemplateEngine* và *TemplateResolver*.
 - Với Spring Boot việc duy nhất cần làm là khai báo 1 dependency của Thymeleaf trong Gradle/ Maven. Spring Boot sẽ tự động cấu hình *ThymeleafView-Resolver*, *SpringTemplateEngine* và *TemplateResolver* cho dự án.

Auto-configuration



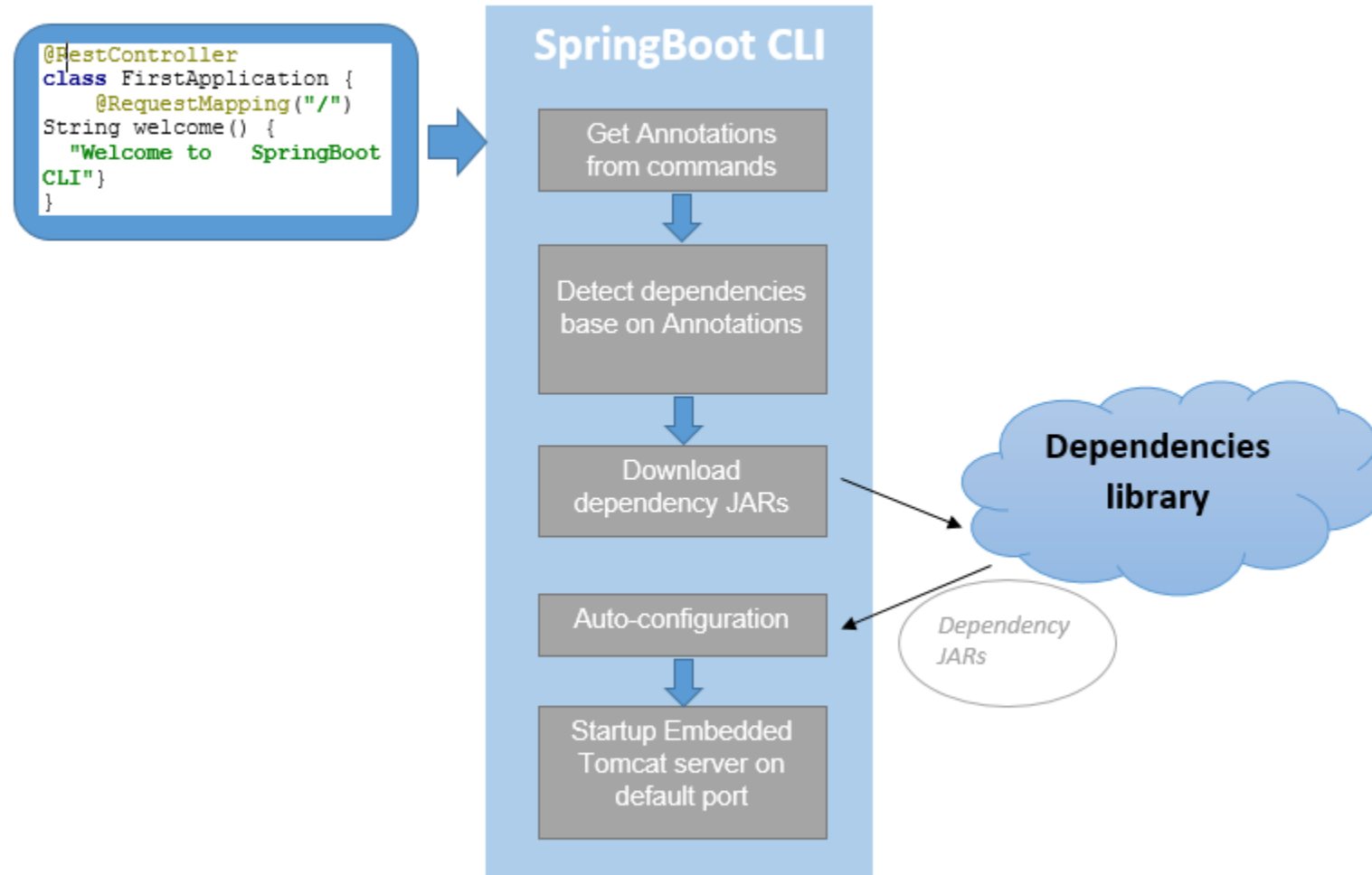
- Ưu điểm:
 - Giảm thiểu rất nhiều khối lượng đội tượng cần cấu hình khi bắt đầu một dự án Spring.
 - Làm đơn giản cấu trúc một dự án Spring giúp người phát triển tập trung nhiều hơn vào phát triển logic nghiệp vụ.
 - Giúp việc chuyển đổi giữa các công nghệ, thư viện dễ dàng hơn.



Command-line interface (CLI)

- SpringBoot-CLI là một giao diện người dùng thao tác qua dòng lệnh hỗ trợ cấu lệnh và cú pháp SpringBoot.
- SpringBoot-CLI tận dụng sự mạnh mẽ của Spring-boot-starter và Auto-configuration để chạy được nhưng dòng lệnh nhập vào mà không cần khai báo dependencies, configuration...

Command-line interface (CLI)



Actuator



- SpringBoot Actuator là một project con nhúng bên trong SpringBoot project.
- SpringBoot Actuator cũng cấp một số dịch vụ giúp theo dõi, kiểm soát hoạt động của SpringBoot Application giống như những publish web services.
 - Quản lý các điểm kết thúc (endpoints):
 - GET /autoconfig: Thông tin về auto-configuration của SpringBoot
 - GET /beans: Danh sách @Bean đang cấu hình trong ứng dụng.
 - GET /env: Lấy danh sách các môi trường và hệ điều hành đang chạy ứng dụng.
 - GET /health: Lấy tình trạng ứng dụng gồm đang hoạt động, đã dừng, dung lượng ổ cứng total, free.
 - POST /shutdown: Cho phép tắt ứng dụng.
 - ...

Actuator



- Gửi thông tin lỗi của ứng dụng.
 - Gửi thông tin tổng quan về quá trình chạy của ứng dụng.
 - Đưa ra framework hỗ trợ kiểm định, kiểm soát các events sinh ra trong quá trình chạy của ứng dụng.
-
- Trong đó tính năng quản lý các điểm kết thúc là quan trọng và hữu ích, hay được sử dụng nhất trong một dự án SpringBoot.

Triển khai SpringBoot

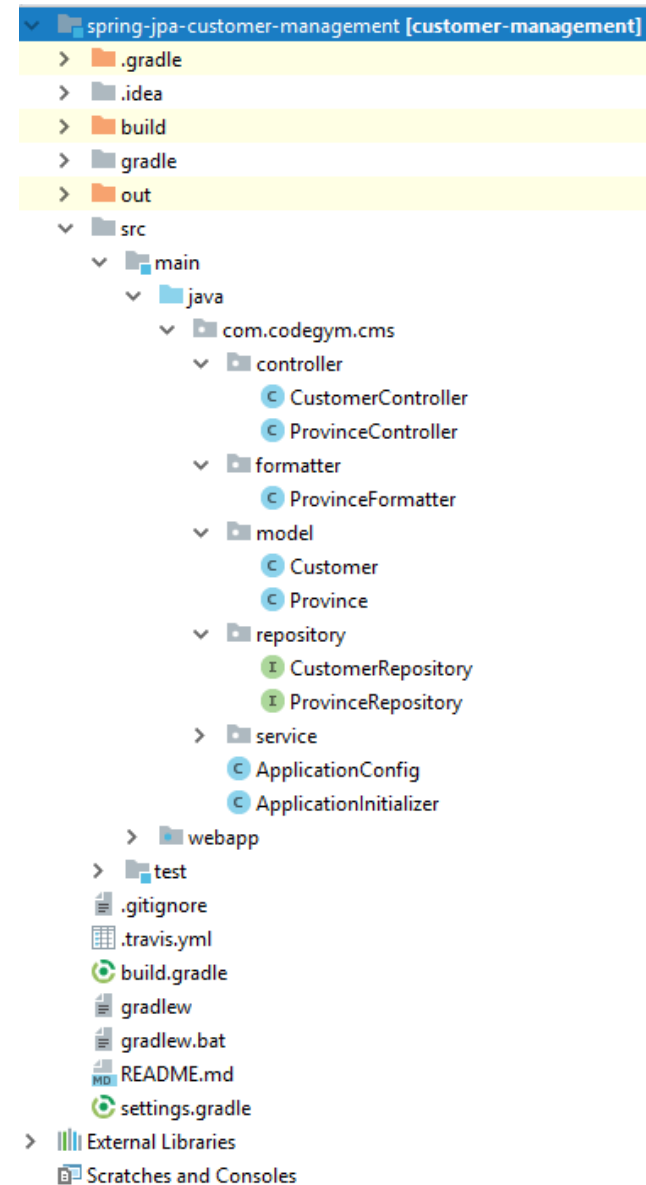


Triển khai Spring Boot (1)

- Hiện nay các công cụ lập trình như IntelliJ, Eclipse...đều hỗ trợ người sử dụng tạo 1 Spring Boot project mang đầy đủ các thuộc tính của Spring Boot hết sức đơn giản.
- Để làm rõ sự khác biệt giữa 1 ứng dụng Spring MVC thông thường, và Spring Boot, tài liệu này sẽ triển khai 1 ứng dụng từ Spring MVC sang Spring Boot.
- Ứng dụng Spring MVC được sử dụng là bài tập đã làm Chương 4.

Triển khai Spring Boot (2)

- Cấu trúc dự án Spring MVC



Triển khai Spring Boot (3)



- Phân tích:
 - Spring Boot với đặc điểm "Spring Boot starters" sẽ sử dụng gói dependency mới để thay thế cho đa số các dependency cũ của Spring MVC nên chắc chắn cấu trúc file Gradle sẽ có thay đổi.
 - Thêm thư viện Spring-boot-gradle-plugin cho dự án, version của gradle plugin sẽ quyết định version của các dependencies khai báo bên dưới.

```
buildscript {  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath ("org.springframework.boot:spring-boot-gradle-plugin:2.0.2.RELEASE")  
    }  
}  
  
group 'com.codegym'  
version '1.0-SNAPSHOT'  
  
apply plugin: 'org.springframework.boot'  
apply plugin: 'io.spring.dependency-management'  
apply plugin: 'java'  
apply plugin: 'jacoco'
```

Triển khai Spring Boot (4)



- Thay thế các dependencies của Spring MVC bằng các dependencies trong gói Spring-boot-starter.

```
dependencies {  
    //-----Spring MVC dependencies -----  
    // compile group: 'org.thymeleaf', name: 'thymeleaf-spring4', version: '3.0.9.RELEASE'  
    // compile group: 'javax.servlet', name: 'javax.servlet-api', version: '3.1.0'  
    // compile group: 'org.springframework', name: 'spring-core', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework', name: 'spring-context', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework', name: 'spring-beans', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework', name: 'spring-web', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework', name: 'spring-orm', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework', name: 'spring-webmvc', version: '4.3.17.RELEASE'  
    // compile group: 'org.springframework.data', name: 'spring-data-jpa', version: '1.11.12.RELEASE'  
    //-----Spring MVC dependencies -----  
  
    //-----Spring Boot Starter dependencies -----  
    compile ("org.springframework.boot:spring-boot-starter-parent")  
    compile ("org.springframework.boot:spring-boot-starter-web")  
    compile ("org.springframework.boot:spring-boot-starter-data-jpa")  
    compile ("org.springframework.boot:spring-boot-starter-thymeleaf")  
    //-----Spring Boot Starter dependencies -----  
}
```

Triển khai Spring Boot (5)



- Phân tích:
 - Với đặc điểm “Auto-configuration” Spring Boot sẽ dựa trên danh sách dependency để tự động cấu hình các thành phần cần thiết cho ứng dụng.
 - Những cấu hình thủ công bên dưới đều có thể bỏ qua:
 - ApplicationInitializer
 - ApplicationContext
 - Thymeleaf configuration (*templateResolver, templateEngine, ThymeleafViewResolver*)
 - JPA configuration
 - Data sources, transactionManager...

Triển khai Spring Boot (6)



- Xóa bỏ 2 class ApplicationInitizlizer và ApplicationConfig, thay vào đó là 1 class Application và hàm Main[] như sau:

```
package com.codegym.cms;

import com.codegym.cms.service.CustomerService;
import com.codegym.cms.service.ProvinceService;
import com.codegym.cms.service.impl.CustomerServiceImpl;
import com.codegym.cms.service.impl.ProvinceServiceImpl;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class Application {

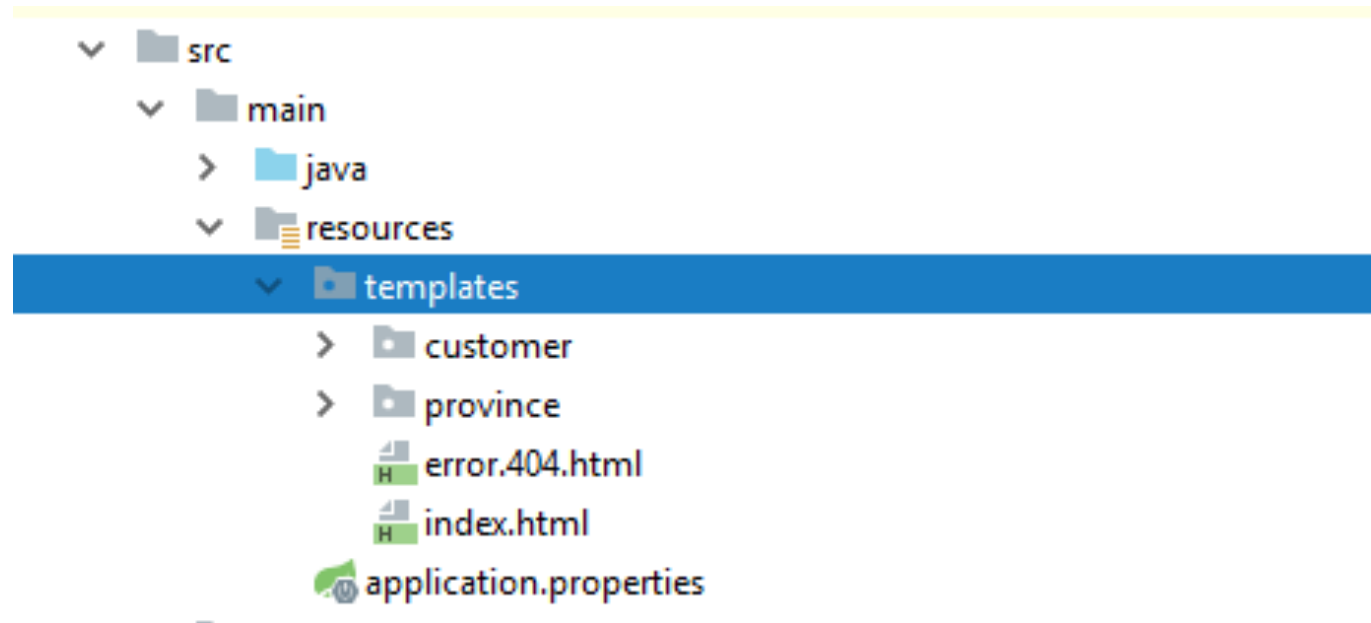
    @Bean
    public CustomerService customerService() { return new CustomerServiceImpl(); }
    @Bean
    public ProvinceService provinceService() { return new ProvinceServiceImpl(); }

    public static void main(String[] args) { SpringApplication.run(Application.class, args); }
}
```


Triển khai Spring Boot (7)



- Phân tích:
 - Spring Boot quản lý các views tại thư mục *src/main/resources/template*.
 - Move tất cả các view trong thư mục *src/main/webapp/WEB-INF/views* sang thư mục template ở trên.



Triển khai Spring Boot (8)



- Phân tích:
 - Spring Boot tự động cấu hình kết nối đến database nhưng vẫn cần cung cấp các thông tin liên quan đến driver, url, username, password.
 - Các thông tin nêu trên được lưu trong file *src/main/resources/application.properties*

```
#----- MySQL database -----  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/cms?useSSL=false  
spring.datasource.username=root  
spring.datasource.password=M0209t2304  
  
#-----Hibernate-----  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=none  
spring.jpa.properties.hibernate.hbm2ddl.auto=update  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

- ```
"C:\Program Files\Java\jdk1.8.0_162\bin\java.exe" ...

 ____ _
 / ___|| | | |
| |___| |_| |
 ___ \| _/
 |_|

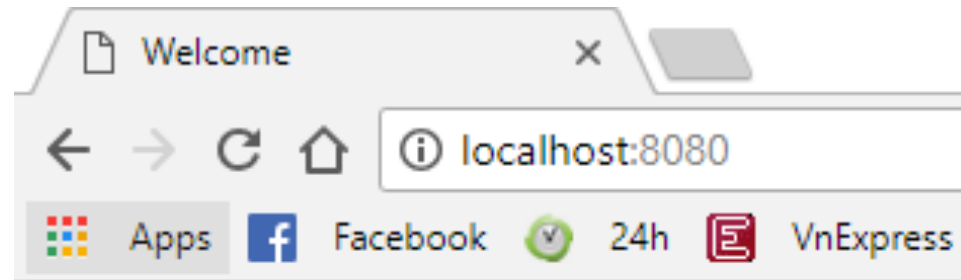
:: Spring Boot :: (v2.0.2.RELEASE)

2018-06-10 20:18:50.873 INFO 1676 --- [main] com.codegym.cms.Application : Starting Application on Dell-M4800 with PID
2018-06-10 20:18:50.876 INFO 1676 --- [main] com.codegym.cms.Application : No active profile set, falling back to defa
2018-06-10 20:18:50.926 INFO 1676 --- [main] ConfigServletWebServerApplication : Refreshing org.springframework.boot.web.ser
2018-06-10 20:18:51.655 INFO 1676 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annot
2018-06-10 20:18:51.855 INFO 1676 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http
2018-06-10 20:18:51.872 INFO 1676 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2018-06-10 20:18:51.873 INFO 1676 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.
2018-06-10 20:18:51.876 INFO 1676 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library
2018-06-10 20:18:51.984 INFO 1676 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplication
2018-06-10 20:18:51.985 INFO 1676 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization
2018-06-10 20:18:52.086 INFO 1676 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2018-06-10 20:18:52.089 INFO 1676 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' t
2018-06-10 20:18:52.090 INFO 1676 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to
2018-06-10 20:18:52.090 INFO 1676 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter'
2018-06-10 20:18:52.090 INFO 1676 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to:
2018-06-10 20:18:52.195 INFO 1676 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2018-06-10 20:18:52.270 INFO 1676 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2018-06-10 20:18:52.300 INFO 1676 --- [main] j.LocalContainerEntityManagerFactoryBean : Building JPA container EntityManagerFactory
2018-06-10 20:18:52.310 INFO 1676 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
```

# Triển khai Spring Boot (9)



- Mở trình duyệt truy cập `http://localhost:8080/`



**Welcome**

[Customers](#) | [Provinces](#)

# Triển khai Spring Boot (10)

- Triển khai Actuator cho 1 dự án Spring Boot

- Thêm dependency cho Actuator.

```
compile ("org.springframework.boot:spring-boot-starter-actuator")
```

- Thêm tham số cấu hình cho Actuator trong application.properties

```
server.port=8080
management.server.port=8090
management.endpoints.web.exposure.include=*
management.endpoint.shutdown.enabled=true
```

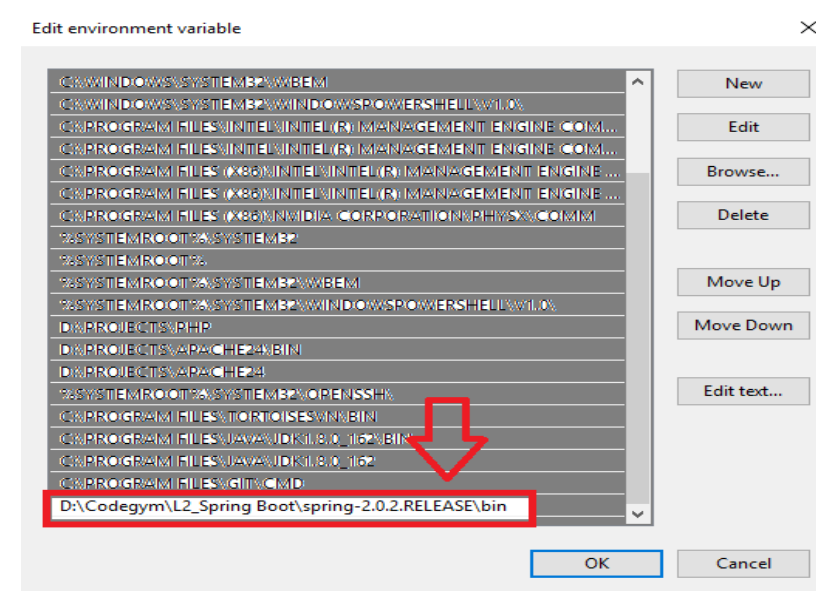
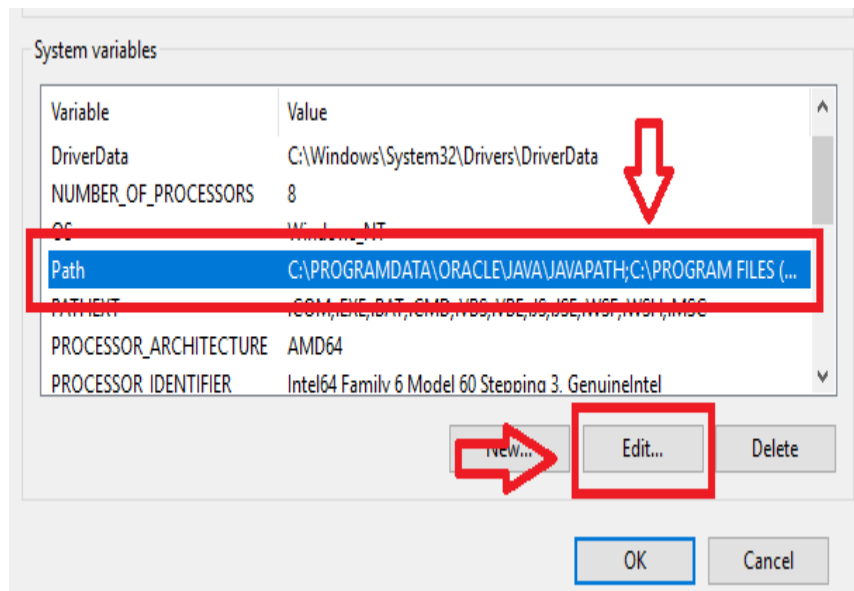
- Chạy ứng dụng, truy cập <http://localhost:8090/actuator> để xem kết quả.
  - Có thể tìm hiểu thêm các endpoint khác như **/actuator/health**, **/actuator/beans**, **/actuator/env** ...

# Spring Boot CLI

# Spring Boot CLI



- Cài đặt Spring Boot CLI (Window)
    - Download Spring Boot CLI tại <http://repo.spring.io/release/org/springframework/boot/spring-boot-cli/>
    - Giải nén file tại thư mục với đường dẫn **PATH** = "... \spring-2.0.2.RELEASE\bin"
    - Thêm đường dẫn PATH vào Environment Variable
- Control Panel -> System -> Advanced system settings -> Environment Variable**



# Spring Boot CLI



- Check trạng thái cài đặt Spring Boot CLI
  - Check version: CMD -> spring --version

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.17134.48]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\M4800>spring --version
Spring CLI v2.0.2.RELEASE
C:\Users\M4800>
```

- Sử dụng help: CMD -> spring --help

```
C:\Users\M4800>spring --help
usage: spring [--help] [--version]
 <command> [<args>]

Available commands are:

run [options] <files> [--] [args]
 Run a spring groovy script

grab
 Download a spring groovy script's dependencies to ./repository

jar [options] <jar-name> <files>
 Create a self-contained executable jar file from a Spring Groovy script

war [options] <war-name> <files>
 Create a self-contained executable war file from a Spring Groovy script

install [options] <coordinates>
 Install dependencies to the lib/ext directory

uninstall [options] <coordinates>
 Uninstall dependencies from the lib/ext directory

init [options] [location]
```



# Spring Boot CLI

---



- Sử dụng Spring Boot CLI
  - Tạo file text với nội dung

```
@RestController
class HelloWorld {
 @RequestMapping("/")
 String welcome() {"Welcome"}
 @RequestMapping("/{name}")
 String hello(@PathVariable("name") String name) {"Hello " + name}}
```

- Đặt tên file là **hello.groovy**



- ```

C:\> Command Prompt

Microsoft Windows [Version 10.0.17134.48]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\M4800>D:

D:\>spring run hello.groovy

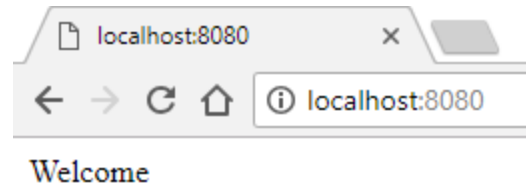
```

- [illegible]

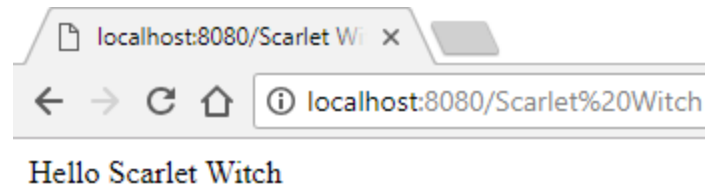
Spring Boot CLI



- Mở trình duyệt truy cập: <http://localhost:8080/>



- Truy cập: `http://localhost:8080/{name}`



Tổng kết



- Spring Boot là một công nghệ mang lại nhiều lợi ích, giúp cho quá trình phát triển một dự án Spring nhanh chóng và dễ dàng hơn rất nhiều.
- Dự án Spring Boot khởi chạy từ hàm `Main[]` nằm trong class được xác định qua annotation `@SpringBootApplication`.
- Một ứng dụng Spring Boot có thể khởi chạy độc lập.