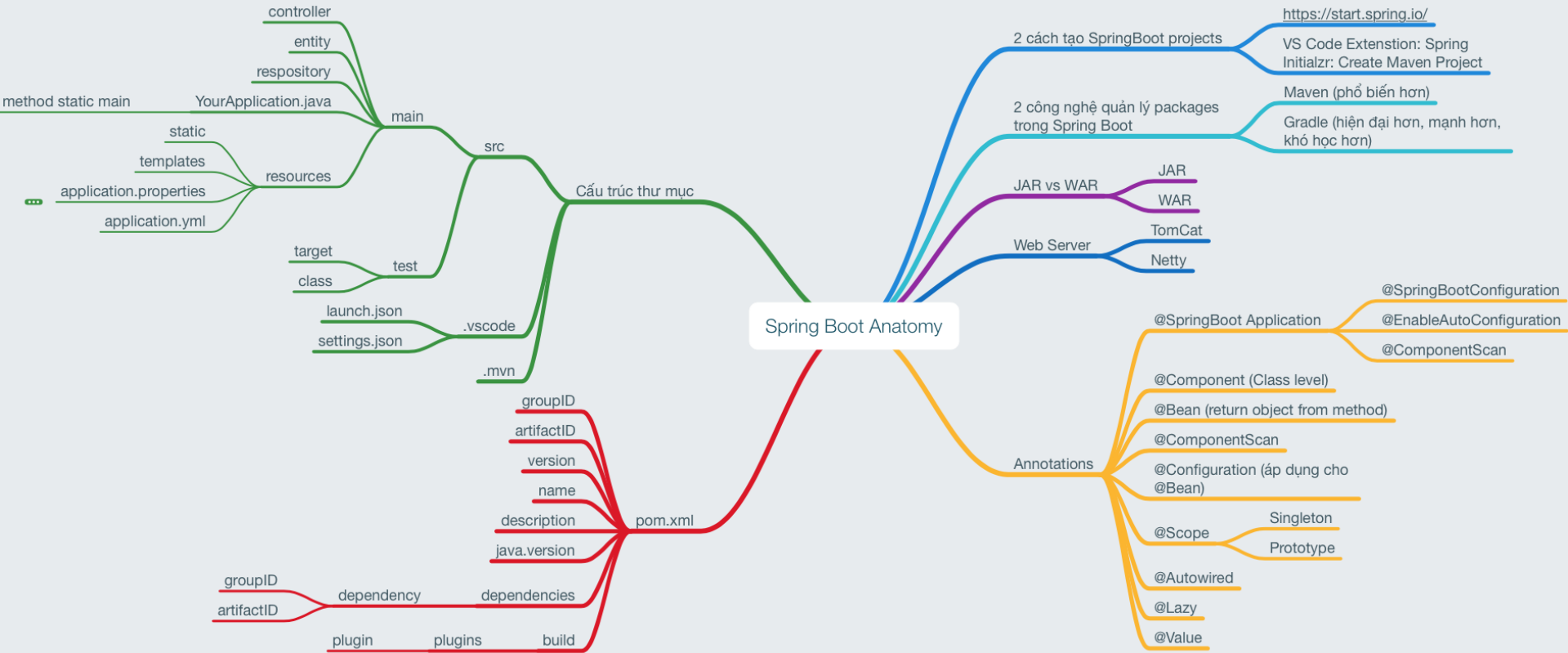


# Cấu trúc dự án SpringBoot

Trịnh Minh Cường

Lập trình viên Techmaster



# Check list

- Có 2 cách tạo Spring Boot project
- Cấu trúc thư mục Spring Boot
- pom.xml
- Class XXXApplication và phương thức static main
- @Controller
- @GetMapping

# Có 2 cách tạo SpringBoot project

1. Sử dụng web site <https://start.spring.io/>



2. Sử dụng extension Spring Initializr Java Support  
[https://marketplace.visualstudio.com/items?vscjava.vscod  
e-spring-initializr](https://marketplace.visualstudio.com/items?vscjava.vscod<br/>e-spring-initializr)

# Ý nghĩa các trường trong project



**Project**

☒ Maven Project ☐ Gradle Project

**Language**

☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**

☐ 2.4.1 (SNAPSHOT) ☒ 2.4.0 ☐ 2.3.7 (SNAPSHOT) ☐ 2.3.6

☐ 2.2.12 (SNAPSHOT) ☐ 2.2.11

## Project Metadata

Group vn.techmaster

Artifact bookstore

Name amazon

Description Online Bookstore

Package name vn.techmaster.bookstore

`<groupId>vn.techmaster</groupId>` //tên miền đảo ngược, reverse domain để phân biệt công ty, tổ chức sở hữu

`<artifactId>bookstore</artifactId>` //tên package chính -> vn.techmaster.bookstore  
//và cũng là tên thư mục dự án

`<version>0.0.1-SNAPSHOT</version>` //phiên bản

`<name>amazon</name>` //tên ứng dụng và tên đầu main class AmazonApplication.java

`<description>Online Bookstore</description>`

pom.xml

Maven và Gradle là 2 công cụ quản lý gói thư viện, build ứng dụng Java

**Maven khác Gradle như thế nào?**

# Maven

- Quản lý thông tin dự án: các thành phần, build, test.. theo mô hình Project object model (POM)
- pom.xml
- Cũ hơn, phổ biến hơn, tuts nhiều hơn
- Sử dụng XML, cấu hình dài dòng
- Sử dụng package trên maven repository

# Gradle

- Quản lý thông tin dự án: build, test...sử dụng Groovy, Kotlin DSL  
Ưu tiên tốc độ và linh hoạt
- build.gradle và settings.gradle
- Mới hơn, chạy nhanh hơn
- Sử dụng script, cấu hình ngắn gọn, thực ra dễ học
- Cũng dùng các package trên maven repository

# Golang Cheat Sheet

## hello.go

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
```

## Run directly

```
$ go run hello.go
Hello, world!
```

## Variables

```
var s1 string
s1 = "Learn Go!"

// declare multiple variables at once
var b, c int = 1, 2
var d = true
```

## Short declaration

```
s1 := "Learn Go!"    // string
b, c := 1, 2         // int
d := true            // bool
```



# Maven pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project ...>
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.0</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>vn.techmaster</groupId>
  <artifactId>bookstore</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>amazon</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>11</java.version>
  </properties>
```

# Maven pom.xml

Code XML rất chi là dài, và dễ gây lỗi

Nhưng vì có rất nhiều dự án mẫu dùng Maven nên chúng ta vẫn phải làm quen với Maven trước

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

</project>
```

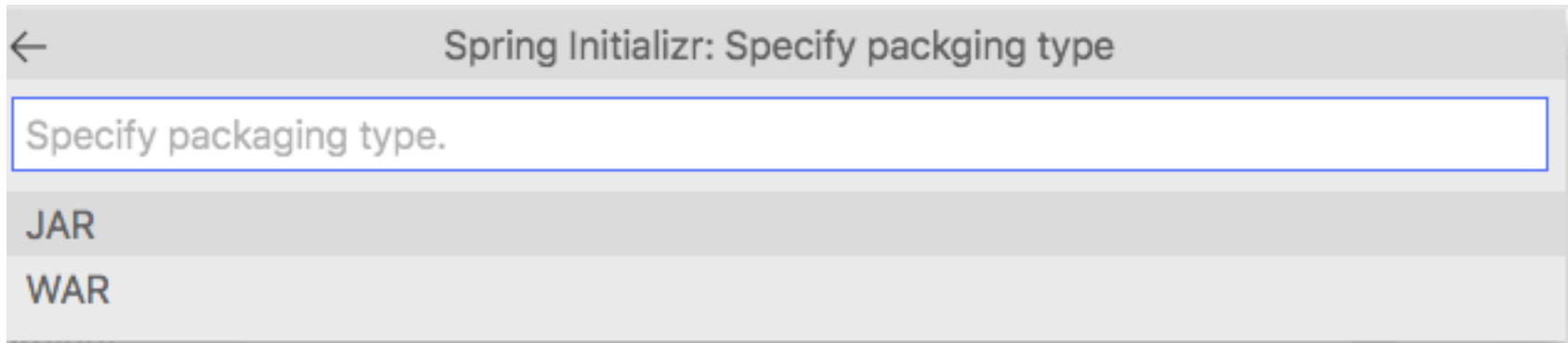
# Gradle build script: build.gradle

```
plugins {  
    id 'org.springframework.boot' version '2.4.0'  
    id 'io.spring.dependency-management' version '1.0.10.RELEASE'  
    id 'java'  
}  
  
group = 'vn.techmaster'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11' //phiên bản JDK  
  
configurations {  
    compileOnly {  
        extendsFrom annotationProcessor  
    }  
}
```

# Gradle build script: build.gradle

```
repositories {  
    mavenCentral()  
}  
  
dependencies { //quản lý các dependencies  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    compileOnly 'org.projectlombok:lombok'  
    developmentOnly 'org.springframework.boot:spring-boot-devtools'  
    annotationProcessor 'org.projectlombok:lombok'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}  
  
test {  
    useJUnitPlatform()  
}
```

Khi dùng Spring Initializr để tạo dự án Spring Boot có màn hình yêu cầu lựa chọn packaging type: JAR hay WAR?

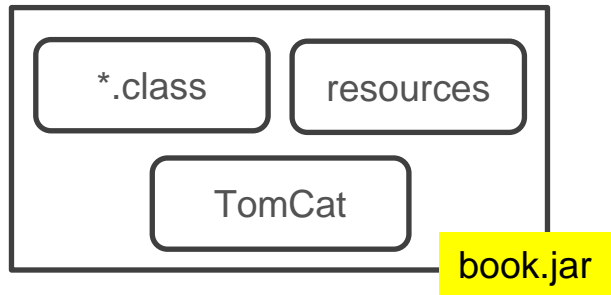


The screenshot shows a dialog box titled "Spring Initializr: Specify packaging type". It features a text input field with the placeholder text "Specify packaging type.". Below the input field are two radio button options: "JAR" and "WAR". The "JAR" option is selected by default.

JAR khác WAR thế nào?

# JAR

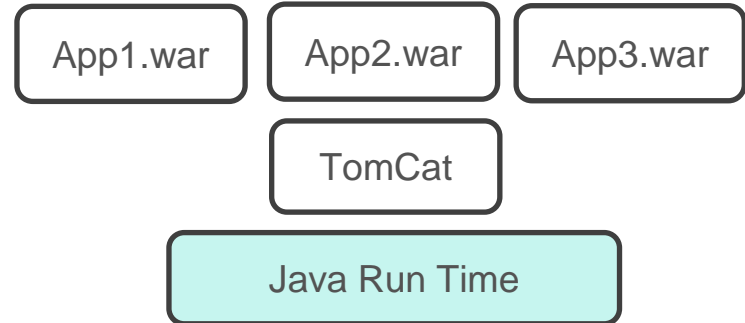
- Java Archive, nén tất cả class và resource lại.
- Chạy JAR file từ command
- Thích hợp để đóng gói trên khai trong Docker image
- Ưu tiên chọn JAR !



\$ java -jar book.jar

# WAR

- Web Application Archive đóng gói web application.
- Cần Web server để chạy WAR
- Phù hợp để chạy nhiều WAR chia sẻ trên cùng một web server.



Spring Boot web app chạy  
web server nào?

Mặc định Spring Boot chạy trên TomCat, web server ổn định, được sử dụng trong nhiều dự án lớn

<http://tomcat.apache.org/>



- TomCat hỗ trợ HTTP/2, WebSocket, NIO (non blocking I/O)
- SpringBoot cũng có thể dùng Netty, một web server thuần sử dụng “non blocking I/O”. Netty mới hơn, chạy nhanh hơn.

Mặc định chúng ta sẽ dùng TomCat vì nó ổn định, tương thích ngược tốt



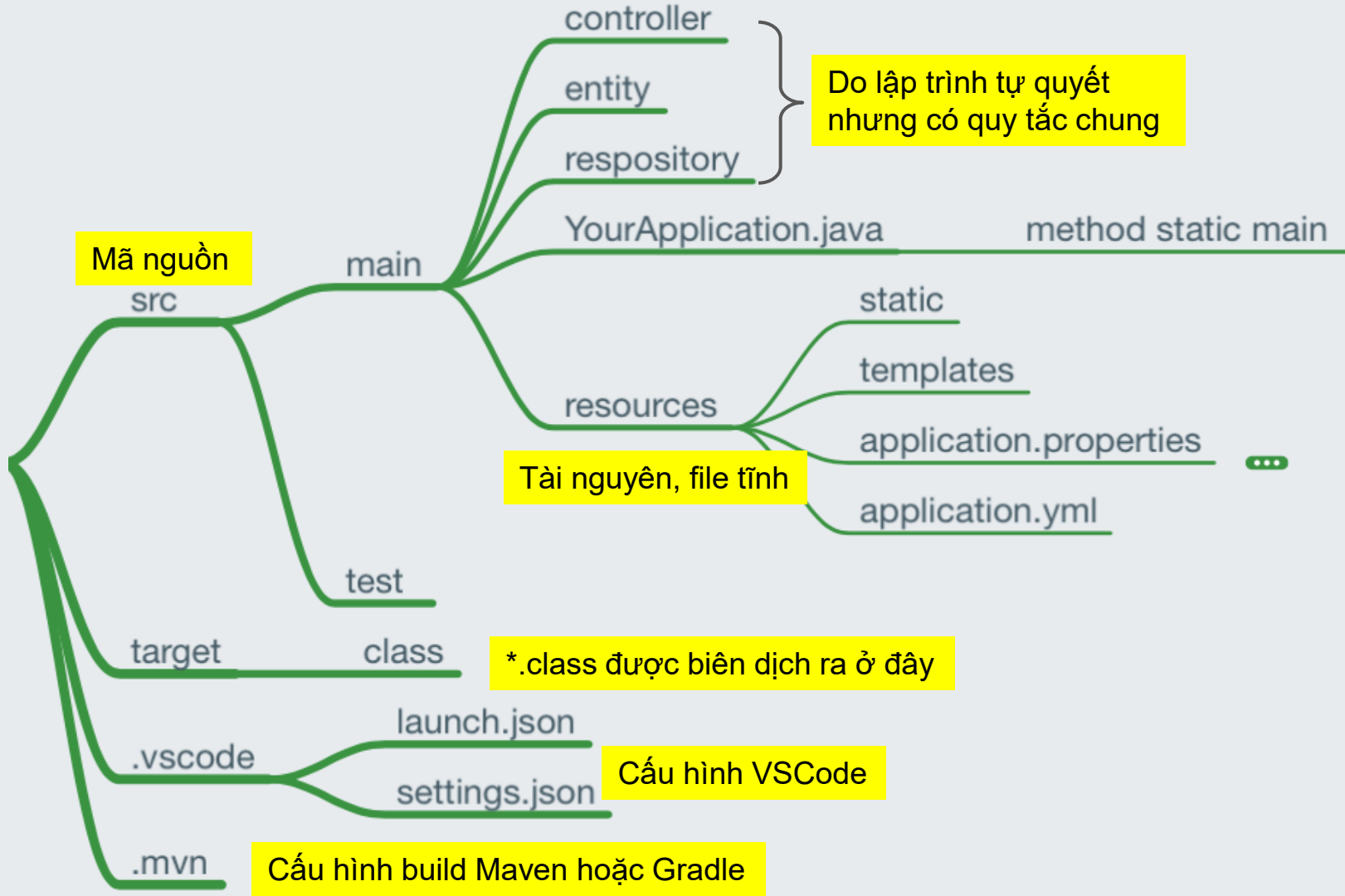
Hãy trình bày cấu trúc thư mục  
một dự án SpringBoot điển hình !

# Cấu trúc thư mục SpringBoot gồm 3 nhóm

1. Mã nguồn cần được quản lý trong **git**
2. Cấu hình VSCode hoặc IntelliJ
3. Cấu hình Gradle hoặc Maven.

Ngoài cùng nên có file

- ReadMe.md mô tả dự án
- .gitignore loại bỏ folders, files không cần quản lý phiên bản



Q: Spring Boot khởi động từ file nào?

A: Từ phương thức static main trong class được đánh dấu **@SpringBootApplication**

# Phương thức static main trong class XApplication

```
package vn.techmaster.bookstore;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
@SpringBootApplication
```

```
public class BookstoreApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(BookstoreApplication.class, args);
```

```
    }
```

Phương thức này đơn giản, hầu như không có code, vậy web app chạy ra sao, sẽ code logic ở đâu?

# Spring Boot startup process

Spring Boot jar file chứa TomCat web server.

Application sẽ tạo mới một process, sau đó khởi động TomCat server

```
[ restartedMain] v.t.w.WorldPopulationApplication      : Starting WorldPopulationApplication on Cuong.local with PID 5060
(/Volumes/CODE/SpringBootHOL/03/WorldPopulation/target/classes started by techmaster in /Volumes/CODE/SpringBootHOL/03/WorldPopulation)
[ restartedMain] v.t.w.WorldPopulationApplication      : No active profile set, falling back to default profiles: default
[ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
[ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.38]
[ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
[ restartedMain] com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Starting...
[ restartedMain] com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Start completed.
[ task-1] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
[ task-1] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.21.Final
[ task-1] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
[ task-1] org.hibernate.dialect.Dialect           : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
[ task-1] o.h.e.t.j.p.i.JtaPlatformInitiator      : HHH000490: Using JtaPlatform implementation:
[org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
[ restartedMain] v.t.w.WorldPopulationApplication      : Started WorldPopulationApplication in 5.893 seconds (JVM running for 7.017)
```

# Bạn sẽ viết code logic ở

- Controller: xử lý các request
- Service: kết nối đến các repository và tương tác với Controller
- Repository: kết nối xuống CSDL qua JPA
- Model: định nghĩa các đối tượng
- Folder Resources/static



Spring Boot web app sử dụng  
những cách nào để cấu hình?

# Có ít nhất 3 cách cấu hình SpringBoot App

1. Các file properties mặc định: application.properties, application.yml
2. Cấu hình bằng annotation: Spring Boot annotations, Lombok annotations hoặc tự tạo ra custom annotation
3. Cấu hình qua file XML

# application.properties

Các thuộc tính nối với nhau qua dấu **chấm** .

```
server.port=8080
```

```
servlet.context-path=/
```

```
spring.main.banner-mode=off
```

```
jpa.database=h2
```

```
jpa.hibernate.dialect=org.hibernate.dialect.H2Dialect
```

```
jpa.hibernate.ddl-auto=create-drop
```

```
freemarker.template-loader-path.classpath=/templates
```

```
freemarker.suffix=.html
```

```
logging.level.org.springframework=ERROR
```

# application.yml

File cấu hình định dạng YAML

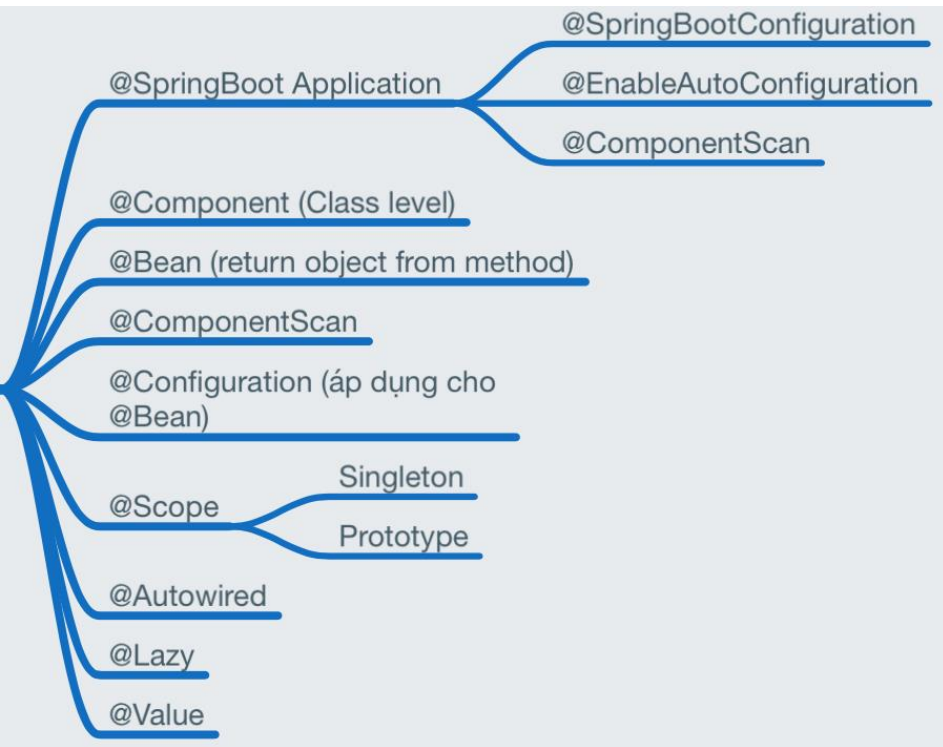
Hàng nào lùi vào một cấp sang phải là thuộc tính con của hàng trên

```
server:
  port: 8080
  servlet:
    context-path: /

spring:
  main:
    banner-mode: "CONSOLE"
  jpa:
    database: h2
    hibernate:
      dialect: org.hibernate.dialect.H2Dialect
      ddl-auto: create-drop
  freemarker:
    template-loader-path: classpath:/templates
    suffix: .html

logging:
  level:
    org:
      springframework: ERROR
```

# Spring Boot Annotations



1. Đánh dấu, phân loại chức năng
2. Tùy chỉnh chức năng thuộc tính và phương thức
3. Thay đổi cách thức Spring Boot hoạt động