



01: Spring Boot

cuong@techmaster.vn

Phương pháp học trong khoá học Spring Boot



Chỉ sau 2.5 tháng bạn sẽ phải đi xin việc

Quy trình tuyển dụng lập trình viên Java SpringBoot:

1. Xét duyệt CV, nếu không có bằng CNTT chính quy một số ngân hàng cổ phần nhà nước sẽ từ chối luôn. Công ty tư nhân vẫn xét tuyển.
2. CV cần liệt kê dự án Java SpringBoot đã tham gia hoặc đề tài tốt nghiệp, link github, bài viết hoặc dịch về SpringBoot
3. Nếu có chứng chỉ Oracle Certified Professional thì càng tốt
4. CV cần có ảnh, trình bày bằng tiếng Việt hoặc Anh dễ đọc, sáng sủa
5. CV có ảnh chụp giao diện web site lập trình bằng SpringBoot thì nhân viên tuyển dụng dễ chú ý

Nhà tuyển dụng sẽ hỏi những gì?

- Kiến thức lập trình Java căn bản, lập trình hướng đối tượng
- Phân tích thiết kế CSDL quan hệ, vững câu lệnh SQL căn bản
- SpringBoot
 - Dependency Injection là gì?
 - Chức năng các Annotation
 - Lập trình REST API
 - POJO, Model, Repository, Service, Controller, RestController
 - Bảo mật REST API JWT, OAuth
 - Truy vấn, cập nhật CSDL: H2, MySQL, Postgresql
 - Xử lý request GET, POST, Upload binary
 - Validate form
 - Đóng gói ứng dụng SpringBoot trong Docker
 - Spring Security ...

Họ có thể yêu cầu bạn code trong 3-4 ngày

- Một REST API service theo một văn bản nghiệp vụ nào đó gồm khoảng 4 bảng, và 12-20 phương thức
- Cần phải xuất văn bản Swagger mô tả lại các phương thức
- Họ thường không yêu cầu layout HTML, CSS đẹp nhưng nhân viên tuyển dụng duyệt hồ sơ thường không hiểu về công nghệ nên nếu có capture màn hình đẹp sẽ thu hút họ xem hồ sơ.
- Sau 3 ngày nếu mới xong khoảng 50% thì có thể xin thêm 1-2 ngày nữa, nếu thực sự muốn xin việc hãy push code lên github có file ReadMe.md giải thích rõ ràng

Họ sẽ hỏi về kinh nghiệm thực tế của các bạn

- Rõ ràng là bạn phải đi làm mới có kinh nghiệm thực tế. Nhưng bạn mới chỉ học 1 tuần 3 buổi ở Techmaster, lấy đâu ra kinh nghiệm thực tế???
- Cách giải quyết là:
 - Hãy bắt đầu lập trình đồ án môn học ngay từ buổi học đầu tiên. Mỗi hôm hoàn thiện một ít, đừng để buổi cuối cùng mới làm, chắc chắn sẽ không xong
 - Điều đặn dịch các bài viết về SpringBoot bằng tiếng Anh, đặt tên bạn lên đó, có cả ảnh và giới thiệu.

Tôi không là giáo viên, tôi là huấn luyện viên

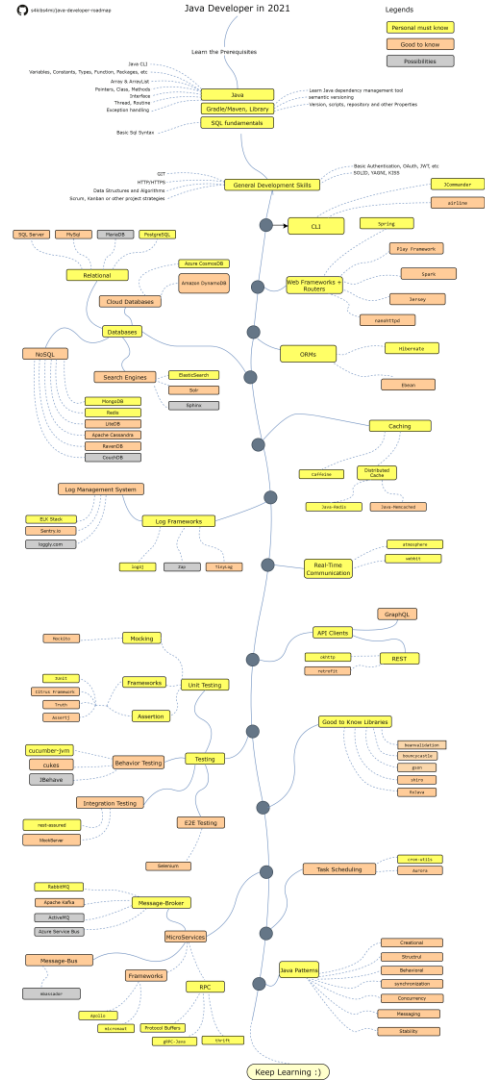
Giáo viên

- Trình bày nhiều, giao bài tập, chữa bài.
- Điểm danh, chấm điểm
- Mục tiêu: Học viên hiểu bài, nhớ bài
- Dạy và dỗ.
- Đáng kính, đáng yêu, trù mẫn

Huấn luyện viên

- Trình bày, chia sẻ sách, tài liệu
- Đề xuất, góp ý ý tưởng dự án
- Hướng dẫn phương pháp học
- Mục tiêu: Học viên làm được dự án, xin được việc kể cả những người kém
- Giống team lead trong đội phần mềm. Có dỗ, nhưng ít thôi, hay tạo áp lực
- Bạn ghét tôi cũng được, nhưng bạn xin được việc !

**Không có nỗ lực nào của bạn
là không được ghi nhận hết.
Do đó hãy nỗ lực đều mỗi
ngày để ra kết quả cụ thể.**



Spring Framework

Co Authors

Juergen Hoeller
Phil Web
Yann Caroff
và cả cộng đồng dev

Các thư viện chính

Spring MVC
Spring Boot
Spring Data
Spring Security
Spring Cloud
Web Flux
Spring Roo
Spring Integration
Spring Flo
Spring Web Flow

Thành tựu

Thư viện lập trình web phổ biến nhất cho ngôn ngữ Java
4th Most Loved Framework in Stack Overflow's 2019

Dòng thời gian

2002 Rod Johnson viết ra mắt bản demo 0.9
2003 Spring 1.0
2006 Spring 2.0
2009 Spring 3.0
2012 đề xuất cải tiến Spring
2013 Spring 4.0
2014 ra mắt SpringBoot (một gói trong Spring Framework)
2017 Spring 5.0
2020 Spring Boot 2.x

Mục tiêu chính

Làm sao lập trình Web bằng Java đơn giản và dễ hơn, bớt sử dụng XML để cấu hình
Quan hệ giữa các thành phần mềm dẻo, linh hoạt, dễ kiểm thử
Đơn giản hoá code của web framework trước đó là Java EE

Ý tưởng chính

Dependency Injection
POJO (Plain Old Java Object)
Convention Over Configuration
Chia thành nhiều thư viện thành phần
Hỗ trợ nhiều 3rd parties, Open Source (Maven, Gradle để cấu hình)

Spring Framework

- Một hệ sinh thái lớn, **một framework Java EE**
- Đơn giản hóa việc phát triển ứng dụng Java EE với mục tiêu loosely coupled bằng cách sử dụng tính năng DI (Dependency injection)
- Cần xác định thủ công các dependencies (file pom, gradle...)
- **Cần cấu hình, cần server để chạy**
- Gồm nhiều **module**: Spring JDBC, Spring MVC, Spring Webflux, Spring Security, Spring AOP, Spring ORM, Spring Test...

SpringBoot

- Chỉ là một **module**
- Nhằm mục đích rút ngắn code và giúp phát triển Web application dễ dàng nhất với tính năng Auto-configuration
- Đi kèm khái niệm starter trong pom, gradle file...giúp xử lý các dependencies dựa trên Spring Boot Requirement
- Gói sẵn các thành phần cần thiết để dev đỡ việc cấu hình phức tạp
- Nhúng web server (Tomcat, Jetty...) vào ứng dụng giúp triển khai dễ dàng hơn

SpringBoot khi so sánh với các web framework khác

- PHP Laravel: cũng là một sinh thái framework, nhiều plugins
- ASP.net Core: Microsoft hỗ trợ,
- Node.js Express: open source, có nhiều module
- Python Django: open source, thiên về CMS, dễ code, nhưng tốc độ thấp

SpringBoot giống gì các web framework khác?

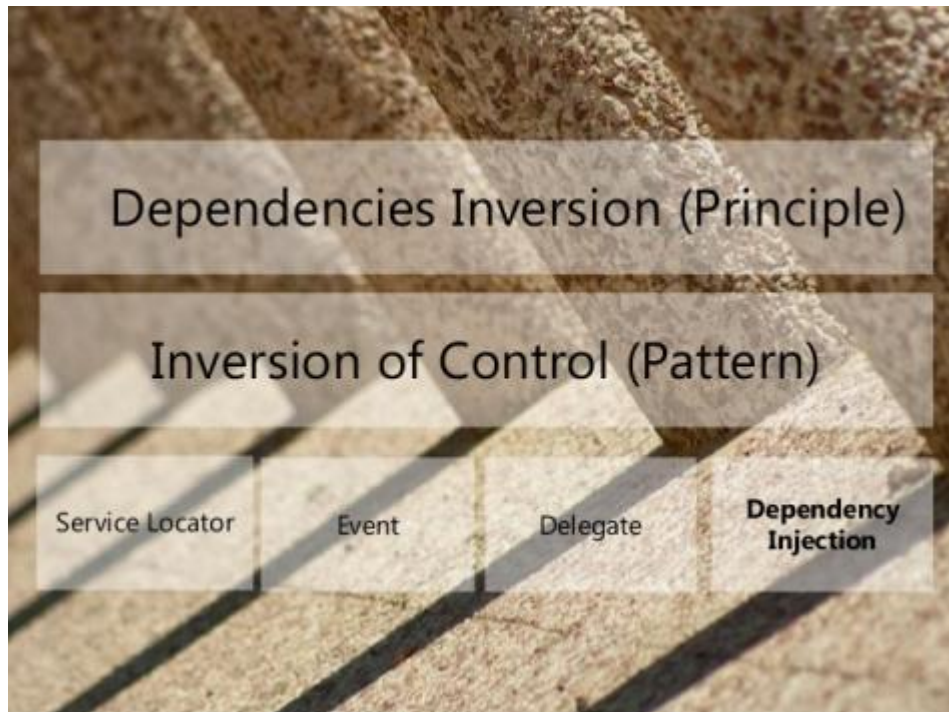
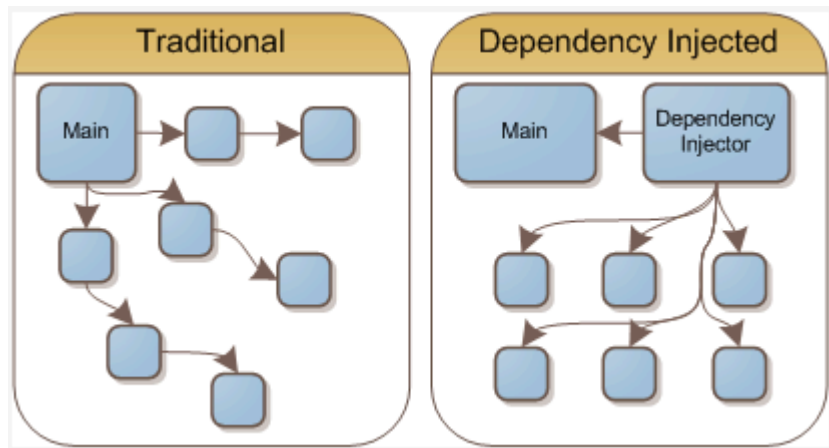
SpringBoot giống các web framework khác ở điểm:

- Xử lý request – response. Bóc tách dữ liệu từ GET path và POST form
- Hỗ trợ REST / JSON or XML. GET, POST, PUT, DELETE..
- Routing: xử lý yêu cầu theo đường dẫn tương ứng
- Session – Cookie
- View Template
- Mô hình Model – View – Controller
- Tương tự middle ware ở Node.js – Express có Interceptor và Filter
- Hỗ trợ nonblock @Async và @EnableAsync

Điểm khác biệt của SpringBoot

- Dependency Injection (DI) và cơ chế quản lý DI (Tính năng của Spring framework):
- Sử dụng **Auto-configuration** với rất nhiều Annotation để đánh dấu, cấu hình, tùy biến class, interface
- **Dễ dàng kết nối và sử dụng các module khác trong Spring framework**
- Để làm gì?
 - Giảm bớt sự phụ thuộc, liên kết cứng nhắc: Tight coupling → Loosely coupling
 - Cấu hình mềm dẻo từ file XML hoặc bằng code @Configuration
 - Dễ lập trình kiểm thử tự động từng thành phần

- IOC và DI



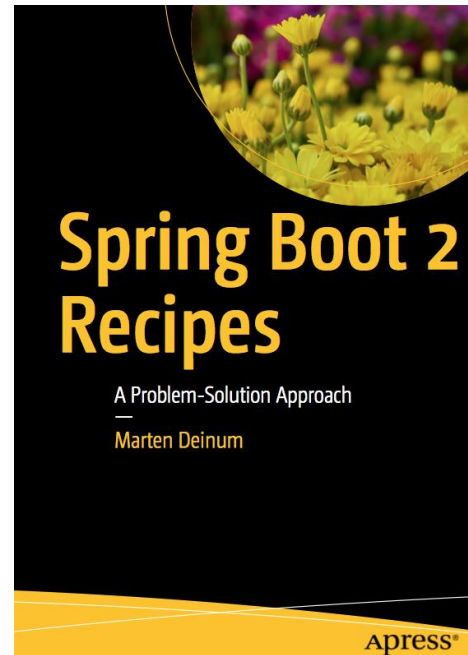
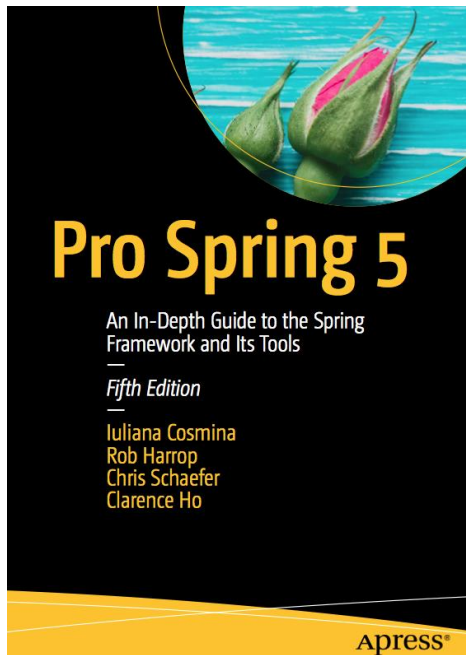
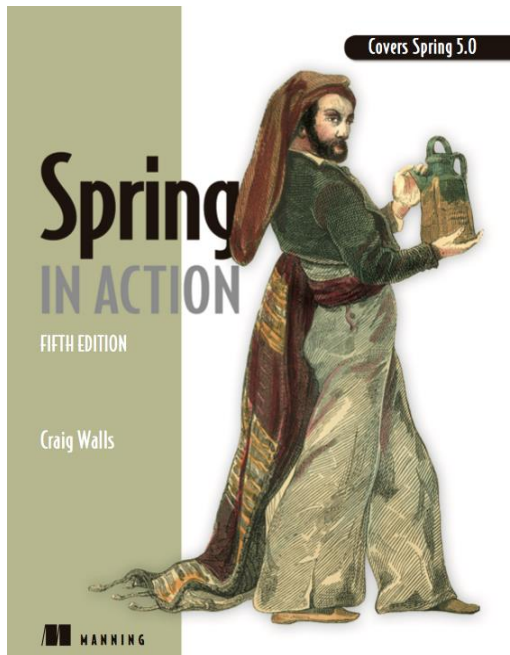
Dành cho những bạn mới học SpringBoot

- 19 bài học căn bản SpringBoot miễn phí tại
<https://loda.me/spring-boot-0-series-lam-chu-spring-boot-zero-to-hero-loda1558963914472/>
- SpringBoot có nhiều các annotation, bài này chú thích đầy đủ
<https://springframework.guru/spring-framework-annotations/>
- Từ điển bách khoa về SpringBoot
<https://www.baeldung.com/>
- Có nhiều code mẫu chạy tốt, giải thích dễ hiểu
<http://zetcode.com/all/#springboot>
- **Doc chính thức của Spring**
<https://spring.io/guides>

Khoá học trên Udemy

- <https://www.udemy.com/course/restful-web-service-with-spring-boot-jpa-and-mysql/>
- <https://www.udemy.com/course/oauth2-in-spring-boot-applications>
- <https://www.udemy.com/course/build-reactive-restful-apis-using-spring-boot-webflux>

Sách hay nên tham khảo một số chương



Phương pháp học

- Sử dụng MindMap để ghi chép, hệ thống hoá lại kiến thức
- Tạo các folder 01-ABC, 02-DEF, 03-MNP, Project cho từng buổi học
- Làm nhiều mini project để thực hành ngay bài vừa học
- Chọn một ý tưởng nhỏ xinh, hợp lý để làm đồ án môn học
- Trao đổi với giảng viên để điều chỉnh ý tưởng thực dụng hơn.
- Lập nhóm 2-3 thành viên cùng trao đổi làm dự án.

Làm việc với giảng viên

- Nếu gặp bug, hãy thử Google hoặc tìm giải pháp trên Stack Overflow, hỏi han bạn bè nếu không tìm ra hãy đẩy code lên github để giảng viên pull về
- Cần mô tả lỗi rõ ràng, các để tạo lại lỗi (reproduce bug)
- Hẹn gặp giảng viên qua email và Zalo, nên tập hợp 2-3 bạn đi cùng để cùng hỏi đáp luôn.
- Thay vì chờ đợi kiến thức từ người khác, hãy tích cực học và chia sẻ những hiểu biết cho mọi người, kể cả giảng viên.

Điều kiện để tốt nghiệp lớp SpringBoot

- Không nghỉ học quá 4 trên tổng số 18 buổi Spring Boot + 6 buổi DevOps
- Bảo vệ đồ án tốt nghiệp SpringBoot, code đẩy lên github, chạy tốt
- Dịch 5 bài viết, hoặc tự viết hướng dẫn lập trình SpringBoot đăng trên blog Techmaster.

Tốt hơn nữa:

- Ôn thi và thi chứng chỉ Oracle Java Professional



01: App đơn giản trả về HTML



Project

2

☒ Maven Project ☐ Gradle Project

3

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 2.4.0 (SNAPSHOT) ☐ 2.4.0 (M3) ☐ 2.3.5 (SNAPSHOT) ☒ 2.3.4
☐ 2.2.11 (SNAPSHOT) ☐ 2.2.10 ☐ 2.1.18 (SNAPSHOT) ☐ 2.1.17

Project Metadata

Group vn.techmaster

Artifact bookstore

Name bookstore 4

Description My book store

Package name vn.techmaster.bookstore

Packaging ☒ Jar ☐ War

Java ☐ 15 ☒ 11 ☐ 8

5

7

Dependencies

ADD DEPENDENCIES... ⌘ + B

Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

6

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf

TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

GENERATE ⌘ + ↵

EXPLORE CTRL + SPACE

SHARE...

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

```
import org.springframework.http.MediaType;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.ResponseBody;
```

```
@Controller
```

```
public class HomeController {
```

```
    @GetMapping(value = "/", produces = MediaType.TEXT_HTML_VALUE)
```

```
    @ResponseBody
```

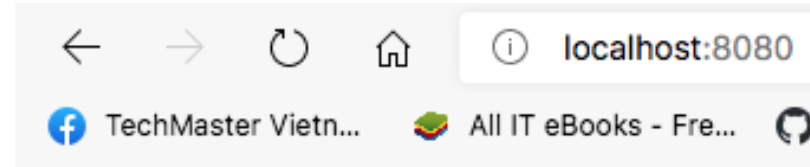
```
    public String home() {
```

```
        return "<h1>Fox Book Store</h1>";
```

```
    }
```

```
}
```

Trả về text response



Fox Book Store

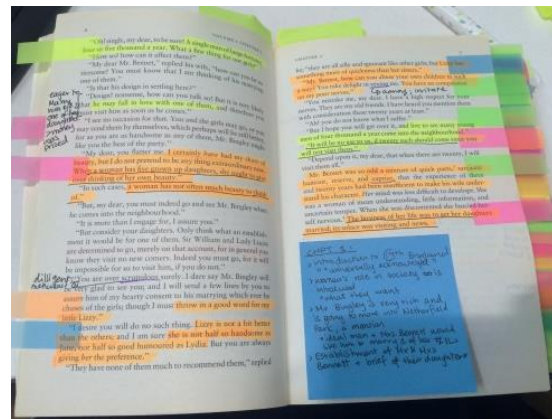
Annotation trong SpringBoot

Annotation để bổ sung thông tin metadata cho class, method, variable.

Metadata này được sử dụng trong quá trình biên dịch (compile time) và cả lúc chạy (run time)

Trong SpringBoot, Annotation dùng để:

1. Đánh dấu phân loại class `@controller`, `@repository`, `@service`
2. Bổ sung thông tin cấu hình ví dụ trong Lombok
3. Bật tắt chức năng nào đó ví dụ như CORS



Annotation – Retention Policy

RetentionPolicy cho biết annotation sẽ được sử dụng đọc ra ở lúc nào

- SOURCE: sẽ bị compiler bỏ qua
 - CLASS: được ghi chú vào class file lúc biên dịch (compile), nhưng bỏ qua khi chạy (run time). Giá trị mặc định
 - RUNTIME: được ghi chú vào class file lúc biên dịch, và đọc ra khi chạy
- C9

TARGET	DESCRIPTION
Annotation Type	Annotates another annotation
Constructor	Annotates a constructor
Field	Annotates a field, such as an instance variable of a class or an <u>enum constant</u>
Local variable	Annotates a local variable
Method	Annotates a method of a class
Module	Annotates a module (new in Java 9)
Package	Annotates a package
Parameter	Annotates a parameter to a method or constructor
Type	Annotates a type, such as a class, interfaces, annotation types, or enum declarations
Type Parameter	Annotates a type parameter, such as those used as formal generic parameters

```
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE) // on class level
public @interface Entity {
    String name();
}
```

Annotation Entity giống với JPA có một tham số (thực ra là phương thức) name

1

```
@Entity(name = "person")
class Person {
}
```

2

Định nghĩa class Person ánh xạ vào bảng person trong CSDL

```
try {
    Entity anno = Person.class.getAnnotation(Entity.class);
    System.out.println("Table name attribute: " + anno.name());
} catch (Exception e) {
    System.out.println(e);
}
```

Lấy giá trị tham số name ra lúc run time

3

Ví dụ phức tạp hơn thôi có nhiều tham số

```
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE) // on class level
public @interface TesterInfo {
    public enum Priority {
        LOW, MEDIUM, HIGH
    }
    Priority priority() default Priority.MEDIUM;
    String[] tags() default "LogicTest";
    String createdBy() default "Techmaster";
    String lastModified() default "2020/12/03";
}
```

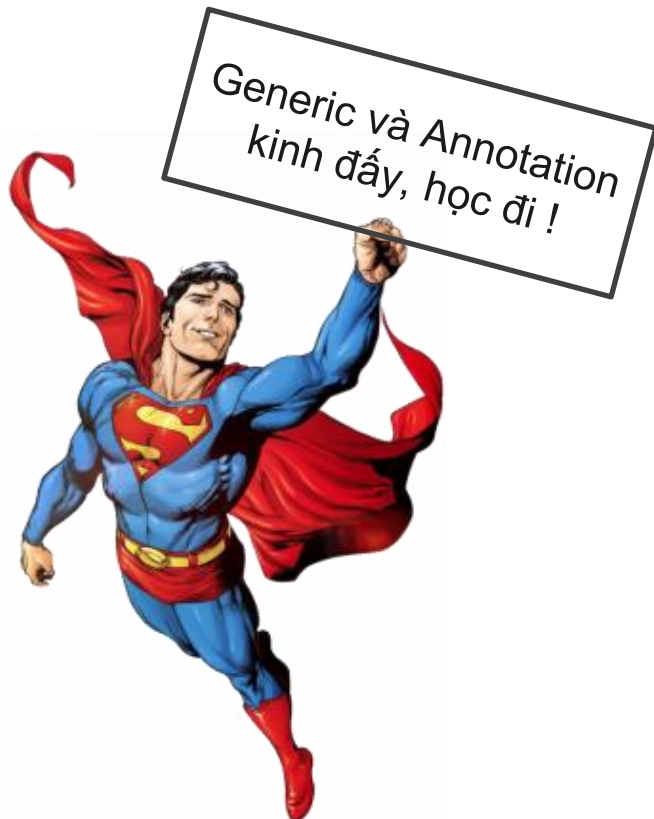
Liệt kê Annotation param bằng reflection

```
public static void listAnnotationParams()
    throws IllegalAccessException, InvocationTargetException {
    for (Annotation annotation : ZooCleaner.class.getAnnotations()) {
        Class<? extends Annotation> type = annotation.annotationType();
        System.out.println("Annotation name: " + type.getName());

        for (Method method : type.getDeclaredMethods()) {
            Object value = method.invoke(annotation, (Object[]) null);
            System.out.println(" " + method.getName() + ": " + value);
        }
    }
}
```

Tại sao phải tìm hiểu kỹ Annotation?

- Bạn có thấy SpringBoot sử dụng rất nhiều Annotation để đánh dấu các class, method, variable không?
- Annotation dùng để cấu hình, bổ sung thông tin phụ trợ cho class, interface, method, variable thay thế cho file cấu hình XML ngoài

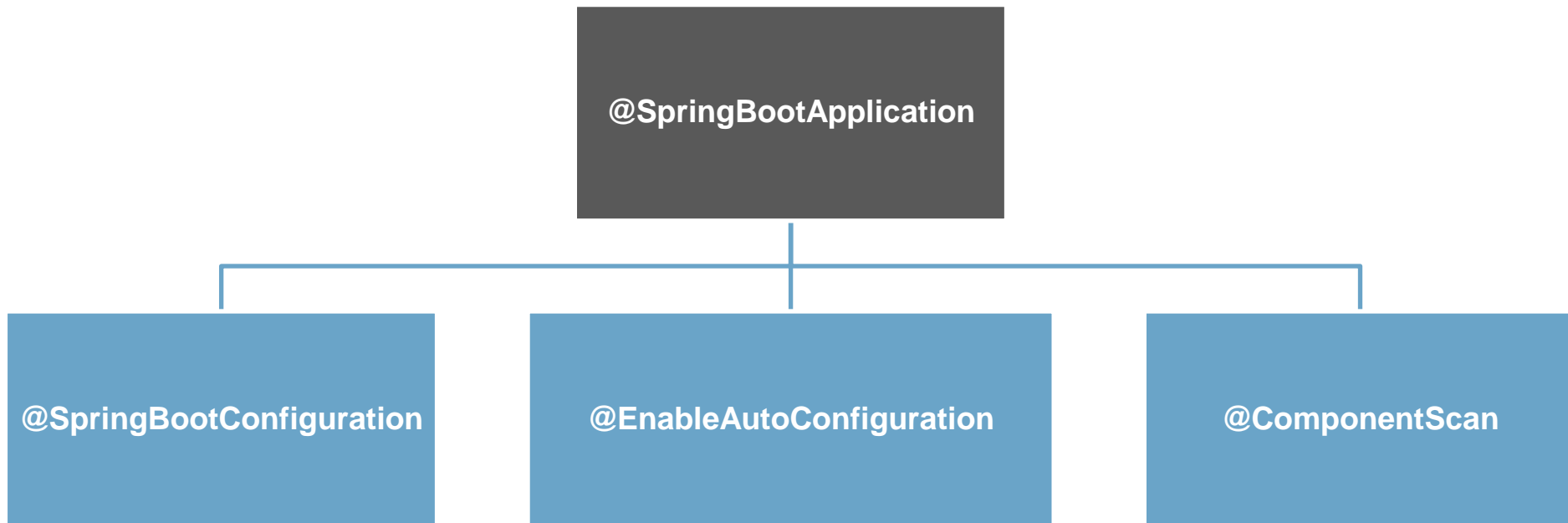


@SpringBootApplication: đây là ứng dụng SpringBoot

@SpringBootConfiguration: cấu hình cho ứng dụng SpringBoot

@EnableAutoConfiguration: cấu hình tự động, bớt thao tác thủ công

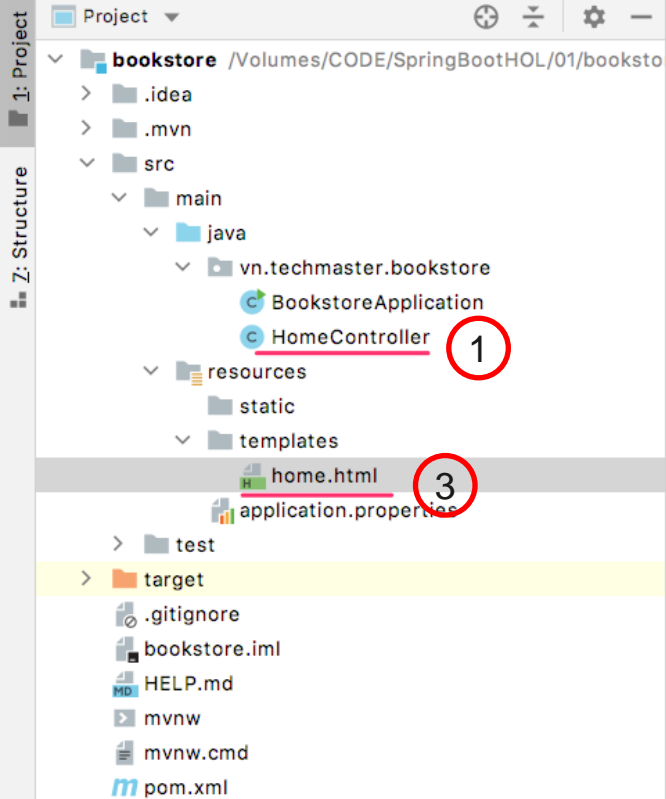
@ComponentScan: quét các component để nạp vào context





02: Sử dụng Thymeleaf

bookstore > src > main > java > vn > techmaster > bookstore > HomeController



HomeController.java x home.html x BookstoreApplication.java x pom.xml (bookstore) x

```
1 package vn.techmaster.bookstore;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5
6 @Controller
7 public class HomeController {
8     @GetMapping("/")
9     public String home() {
10         return "home"; (2)
11     }
12 }
```

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

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
```

```
public class HomeController {
```

```
    @GetMapping("/")
```

```
    public String home() {
```

```
        return "home";
```

```
    }
```

```
}
```

bookstore > src > main > resources > templates > home.html

Project



home.html



1: Project

Z: Structure

- bookstore /Volumes/CODE/SpringBootHOL/01/booksto
 - .idea
 - .mvn
 - src
 - main
 - java
 - resources
 - static
 - templates
 - home.html
 - application.properties
 - test

```
1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <title>Fox Book Store</title>
6      </head>
7      <body>
8          <h1>Book Store</h1>
9          <body>
10     </html>
```

```
<!DOCTYPE html>
<html lang="en">
<headmeta charset="UTF-8">
  <titletitle>
</head>
<bodyh1h1>
</body>
</html>
```



03: Xử lý lỗi

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

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
```

```
public class HomeController {
```

```
    @GetMapping("/about")
```

```
    public String home() {
```

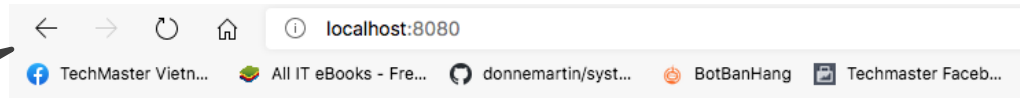
```
        return "home";
```

```
    }
```

```
}
```

1. Thay ("/") bằng ("/about")

2. Vẫn truy cập localhost:8080



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Thu Oct 08 17:14:04 ICT 2020

There was an unexpected error (type=Not Found, status=404).

No message available

Nếu gặp lỗi này xử lý thế nào?

← → ↻ 🏠 ⓘ localhost:8080/home

📘 TechMaster Vietn... 📖 All IT eBooks - Fre... 🔄 donnemartin/syst... 🍷 BotBanHang 📺 Techmaster Faceb..

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Thu Oct 08 16:06:50 ICT 2020

There was an unexpected error (type=Not Found, status=404).

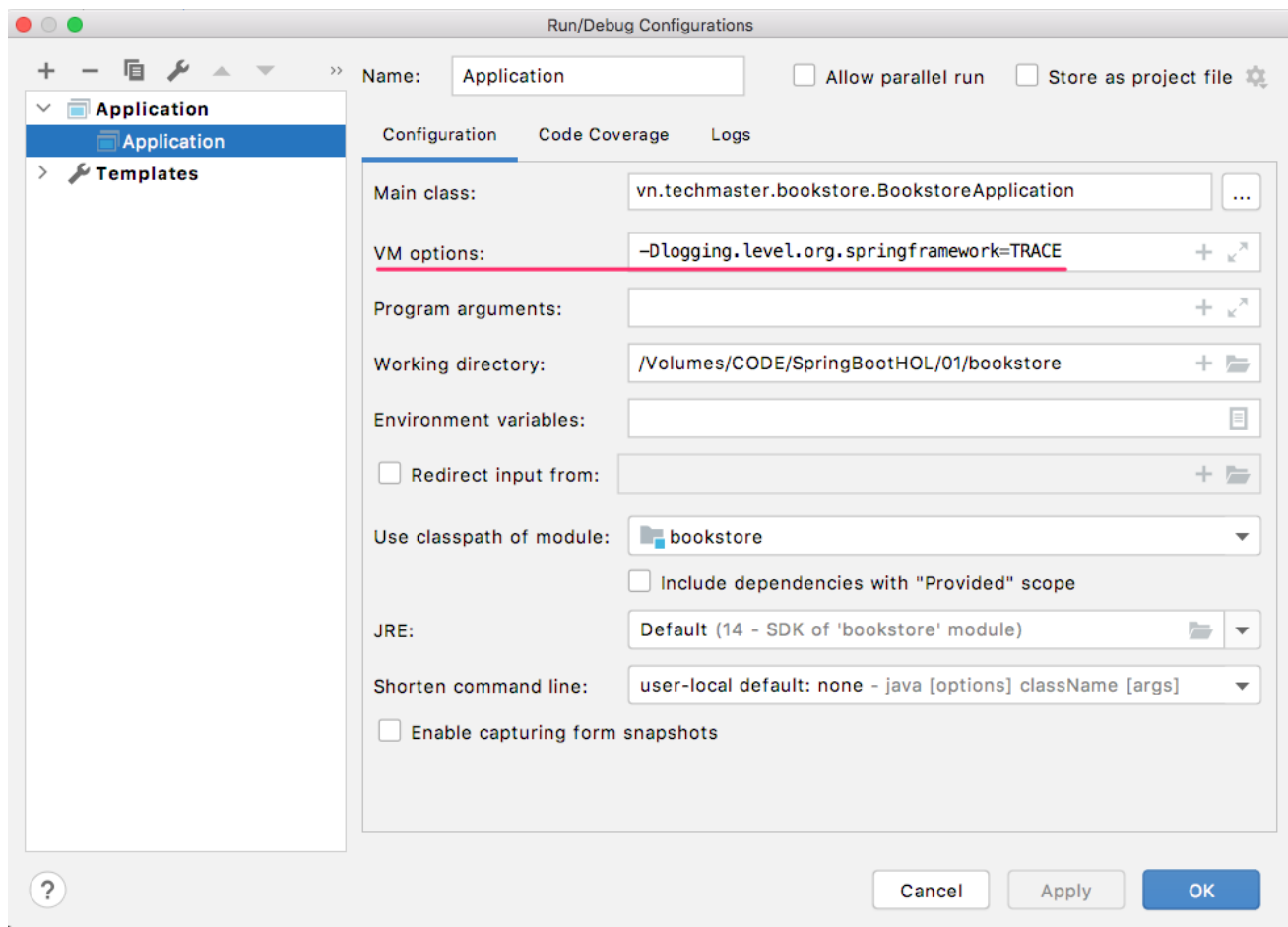
No message available

Nguyên nhân gây lỗi

1. Không có dependency Thymeleaf trong pom.xml
2. Sai đường dẫn ở annotation @GetMapping
3. Tên file *.html trong template bị sai

Bổ sung Logging để nhìn lỗi kỹ hơn

-`Dlogging.level.org.springframework=TRACE`



Nhìn vào log thấy được nhiều thông tin bổ ích tìm ra lỗi ứng dụng

The screenshot displays an IDE interface with two main panels. The top panel shows the project structure on the left and the code editor on the right. The code editor is open to `home.html` in the `resources/templates` directory. The code contains an HTML structure with a head, body, and h1 tag. The h1 tag on line 8 is highlighted in blue and contains the text "Có gì hay nào". The bottom panel shows the application's log output, which details the request processing from the dispatcher servlet to the controller and back to the view.

bookstore > src > main > resources > templates > home.html

Project > bookstore /Volumes/CODE/SpringBootHOL/01/bookstore

home.html x

```
6 </head>
7 <body>
8 <h1>Có gì hay nào</h1>
9 </body>
10 </html>
```

html > body > h1

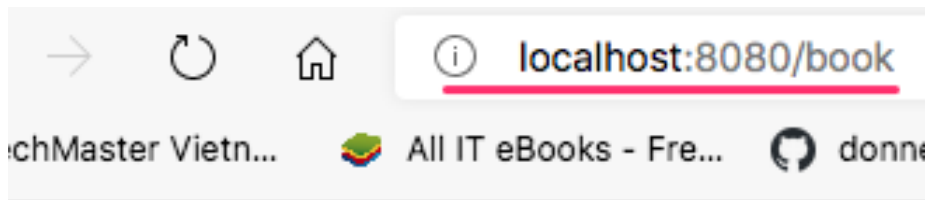
Run: Application x

Bound request context to thread: org.apache.catalina.connector.RequestFacade@31d0e1f8
GET "/", parameters={}, headers={masked} in DispatcherServlet 'dispatcherServlet'
Returning cached instance of singleton bean 'homeController'
Mapped to vn.techmaster.bookstore.HomeController#home()
Arguments: []
View with key [home] served from cache
Selected 'text/html' given [text/html, application/xhtml+xml, image/webp, image/apng, application/xml;q=0.9, applicat
Rendering view [org.thymeleaf.spring5.view.ThymeleafView@536e1ec7]
Completed 200 OK, headers={masked}
Cleared thread-bound request context: org.apache.catalina.connector.RequestFacade@31d0e1f8



04: Trả về dữ liệu cho Thymeleaf

Cần trả về danh sách các quyển sách



- Deep Work
- Nhà Giả Kim
- Cafe cùng Tony
- Tôi đi code dạo

```
package vn.techmaster.bookstore;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;
```

@Controller

```
public class HomeController {
```

```
    @GetMapping("/book")
```

```
    public String book(Model model) {
```

```
        String[] bookCollection = {
```

```
            "Deep Work", "Nhà Giả Kim ", "Cafe cùng Tony", "Tôi đi code dạo"
```

```
        };
```

```
        model.addAttribute("books", bookCollection);
```

```
        return "book";
```

```
    }
```

```
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Book</title>
</head>
<body>
<ul th:each="book : ${books}">
  <li th:text="${book}"></li>
</ul>
</body>
</html>
```

```
<ul>
  <li>Deep Work</li>
</ul>
<ul>
  <li>Nhà Giả Kim </li>
</ul>
<ul>
  <li>Cafe cùng Tony</li>
</ul>
<ul>
  <li>Tôi đi code dạo</li>
</ul>
```




05: Hot (live) reload (Hot Swap)
– làm mới web site
mà không cần biên dịch lại

Hot (live) Reload là gì?


- Live Reload sẽ tự động cập nhật lại giao diện ứng dụng khi lập trình viên sửa đổi code mà không cần phải biên dịch lại toàn bộ ứng dụng.
- Với ngôn ngữ Python, JavaScript, PHP, Ruby việc live reload ứng dụng web rất đơn giản.
- Với C#, Golang, Java hot reload là một kỹ thuật phức tạp.
- SpringBoot **chỉ live reload khi code xuất giao diện hay những file trong thư mục resources thay đổi**. Nếu dự án bổ sung thêm thư viện mới, live reload sẽ không còn tác dụng, cần phải stop và build lại dự án.

Dùng IDE để hot reload dễ nhất

- Visual Studio Code hỗ trợ hot reload với Edge rất tốt, hoạt động trơn tru.
- IntelliJ bản Community không hỗ trợ hot reload (mặc định), cần cấu hình thêm, mà chỉ có bản Ultimate.
- Eclipse STS

Thực hành

- Dùng Visual Studio Code và IntelliJ Community để thử chức năng Hot Reload
- Thay đổi nội dung file about.html
- Kiểm tra web site tại <http://localhost:8080/about>



```
<!DOCTYPE html>
<html lang="en">

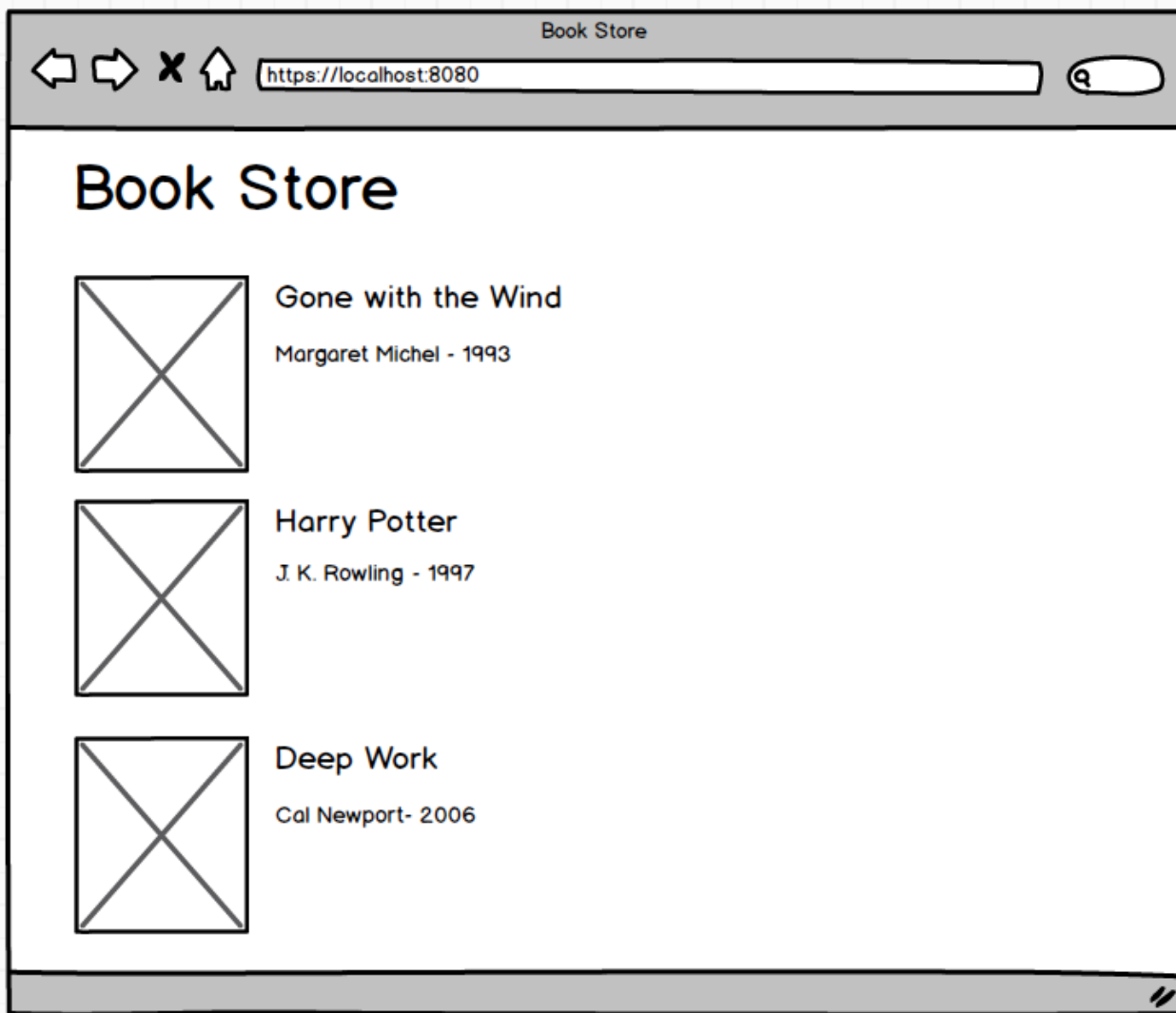
<head>
<meta charset="UTF-8">
<title>About</title>
</head>

<body>

<br>
<p>Hãy thay đổi ở đây</p>
</body>
</html>
```



01: Đổ dữ liệu vào Thymeleaf



Hãy xây dựng một web site có trang chủ gồm ảnh bìa sách và tiêu đề

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

```
public class Book {  
    int id;  
    String title;  
    String coverPhoto;
```

```
    Book(int id, String title, String coverPhoto) {  
        this.id = id;  
        this.title = title;  
        this.coverPhoto = coverPhoto;  
    }
```

```
// Nếu không có getter/setter thì sẽ lỗi ngay đây.
```

```
    public long getId() {  
        return id;  
    }
```

```
    public void setId(int id) {  
        this.id = id;  
    }
```

```
    public String getTitle() {  
        return title;  
    }
```

```
    public void setTitle(String title) {  
        this.title = title;  
    }  
}
```

1. Định nghĩa class Book

2. Phải có getter nếu muốn lấy dữ liệu ở Thymeleaf View

@Controller

```
public class HomeController {  
    @GetMapping("/")  
    public String book(Model model) {
```

```
        ArrayList<Book> books = new ArrayList<>();
```

```
        books.add(new Book(1, "Good with the wind", "gone_with_the_wind.jpg"));
```

```
        books.add(new Book(2, "Nhà Giả Kim Thuật", "gia_kim_thuat.jpg"));
```

```
        books.add(new Book(3, "Harry Porter", "harry_porter.jpg"));
```

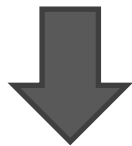
```
        books.add(new Book(4, "Deep Work", "deep_work.jpg"));
```

```
        model.addAttribute("books", books);
```

```
        return "book";
```

```
    }
```

```
}
```



```
<h1>Book Store</h1>
```

```
<ul th:each="book : ${books}">
```

```
<li th:text="${book.id} + ' - ' + ${book.title}"></li>
```

```
</ul>
```

HomeController.java

book.html