**Bạn có thật sự hiểu rõ @Configuration annotation trong Spring**

Spring *@Configuration* là một trong những thành phần cốt lõi của Spring framework, nó đại diện cho một class dùng để sản sinh các [bean](https://shareprogramming.net/tai-sao-spring-bean-duoc-goi-la-xuong-song-cua-ung-dung/) thông qua các method được định nghĩa bên trong. Spring sẽ quét những class này để xử lý và khởi tạo các bean dùng trong toàn bộ ứng dụng.

Trong bài viết này chúng ta sẽ cùng nhau tìm hiểu về cách sử dụng *@Configuration* annotation, một trong những annotation cốt lõi của Spring mà bạn phải tìm hiểu đầu tiên trước khi muốn tiến xa hơn với Spring hay Spring Boot Framework.

**@Configuration annotation**

Như đã thảo luận ở trên *@Configuration* annotation cho phép chúng ta khai báo một class dùng để khởi tạo các bean trong Spring. Trước tiên mình có một class *MyBean* như sau:

package com.deft.configuration;

public class MyBean {

public MyBean(){

System.out.println("My Bean is created");

}

}

Sau đó, mình dùng *@Configuration* để khởi tạo một class trong đó mình sẽ tạo một bean có kiểu dữ liệu là *MyBean* mà mình đã định nghĩa ở trên.

package com.deft.configuration;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class MyConfig {

@Bean

public MyBean myBean() {

return new MyBean();

}

}

Trong bài viết này, mình sẽ sử dụng [Spring Boot](https://shareprogramming.net/spring-boot-la-gi-no-khac-gi-so-voi-spring/) để khởi chạy ứng dụng Spring nhầm đơn giản hoá quá trình cấu hình, giúp chúng ta chú tâm vào *@Configuration* hơn là framework sử dụng. Vì cho dù Spring hay Spring Boot thì *@Configuration*đều sử dụng được và có tính năng tương tự.

package com.deft.configuration;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class ConfigurationApplication {

public static void main(String[] args) {

SpringApplication.run(ConfigurationApplication.class, args);

}

}

Mặc định, *@SpringBootApplication* đã bao gồm các tính năng của các annotation ***@Configuration, @EnableAutoConfiguration***và ***@ComponentScan*** sử dụng để khởi chạy một ứng dụng Spring.

Kết quả sau khi mình khởi chạy ứng dụng các bạn có thể thấy màn hình console sẽ xuất ra **“My Bean is created”**. Như vậy chúng ta có thể kết luận rằng *MyBean* đã được Spring khởi tạo và đăng ký với IoC.

2020-12-20 16:10:25.297 INFO 2580 --- [ main] c.d.c.ConfigurationApplication : Starting ConfigurationApplication using Java 1.8.0\_271 on Buis-MacBook-Pro.local with PID 2580 (/Users/kyle/Desktop/spring\_tutorials/configuration/target/classes started by kyle in /Users/kyle/Desktop/spring\_tutorials/configuration)

2020-12-20 16:10:25.301 INFO 2580 --- [ main] c.d.c.ConfigurationApplication : No active profile set, falling back to default profiles: default

My Bean is created

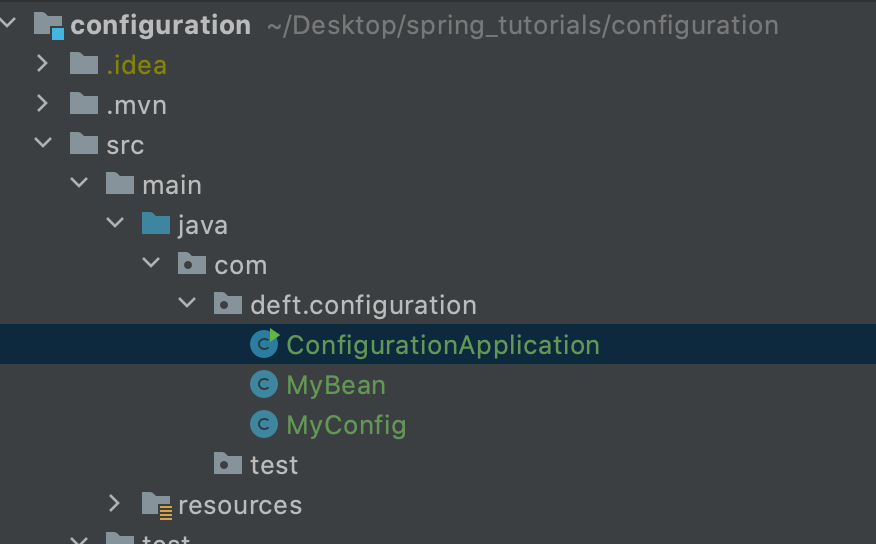
2020-12-20 16:10:26.191 INFO 2580 --- [ main] c.d.c.ConfigurationApplication : Started ConfigurationApplication in 1.984 seconds (JVM running for 3.085)

**@SpringBootApplication**

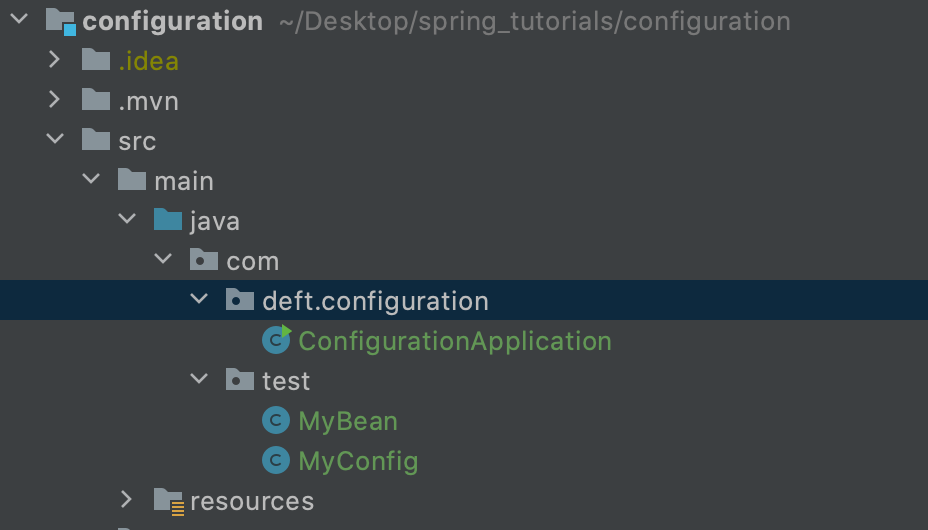
Ở phần trình bày trên thì có lẽ ai cũng đã hiểu được mục đích của *@Configuration* là dùng để định nghĩa bean thông qua các method được định nghĩa bên trong rồi phải không nào. Tuy nhiên một điểm quan trọng mà chúng ta cần làm rõ để có thể hiểu sâu hơn về *@Configuration*.

*@SpringBootApplication* bản thân nó là sự kết hợp các tính năng của ***@Configuration, @EnableAutoConfiguration***và ***@ComponentScan.***Ngoài việc giúp Spring Boot tự động cấu hình và khởi chạy ứng dụng thì [*@ComponentScan*](https://shareprogramming.net/quet-component-trong-spring-boot-voi-componentscan/) được dùng để chỉ định những nơi mà Spring nên tìm kiếm các bean để khởi tạo và đăng ký với IoC.

*@SpringBootApplication*sử dụng *@ComponentScan*mặc định không có tham số, lúc này Spring sẽ tìm kiếm các bean tại cùng package và các package con của nó. Chúng ta có thể thấy rằng *MyBean, MyConfig*, và *Main Class – ConfigurationApplication*đều nằm cùng package *com.deft.configuration*nên khi ứng dụng chạy lên nó Spring có thể quét và khởi tạo bean trong *MyConfig*.

[](https://shareprogramming.net/wp-content/uploads/2020/12/Screen-Shot-2020-12-20-at-4.24.18-PM.png)

Giả sử như mình chuyển *MyBean* và *MyConfig* sao packagepackage com.test nghĩa là chúng không cùng package với *ConfigurationApplication*cũng không phải là package con của nó.

[](https://shareprogramming.net/wp-content/uploads/2020/12/Screen-Shot-2020-12-20-at-4.26.34-PM.png)

Thì khi mình khởi chạy ứng dụng lên

2020-12-20 16:26:21.086 INFO 2661 --- [ main] c.d.c.ConfigurationApplication : Starting ConfigurationApplication using Java 1.8.0\_271 on Buis-MacBook-Pro.local with PID 2661 (/Users/kyle/Desktop/spring\_tutorials/configuration/target/classes started by kyle in /Users/kyle/Desktop/spring\_tutorials/configuration)

2020-12-20 16:26:21.091 INFO 2661 --- [ main] c.d.c.ConfigurationApplication : No active profile set, falling back to default profiles: default

2020-12-20 16:26:21.705 INFO 2661 --- [ main] c.d.c.ConfigurationApplication : Started ConfigurationApplication in 1.387 seconds (JVM running for 1.869)

Chúng ta có thể thấy rằng không có một dòng chữ **“My Bean is created”**nào được in ra màn hình console. Trong trường hợp này, bắt buộc chúng ta phải override lại *@ComponentScan* để chỉ ra những nơi Spring nên quét các bean, mà trong trường hợp của chúng ta là *com.test*.

@SpringBootApplication

@ComponentScan(basePackages = {"com.test"})

public class ConfigurationApplication {

public static void main(String[] args) {

SpringApplication.run(ConfigurationApplication.class, args);

}

}

**Kết bài**

Hy vọng bài viết này sẽ giúp bạn hiểu rõ hơn về *@Configuration*và hãy nhớ rằng, nếu sử @*Configuration* nằm khác package với Main class trong ứng dụng spring, thì hãy chỉ định đường dẫn trong *@ComponentScan* để spring có thể nhận biết và khởi tạo chúng nhé.

Sau cùng, các bạn có thể tham khảo mã nguồn được mình công khai trên gitlab để tiện thực hành: [configuration](https://github.com/devjoyvn/spring_tutorials/tree/master/configuration)

https://shareprogramming.net/ban-co-that-su-hieu-ro-configuration-annotation-trong-spring/