**Spring Java Base Configuration**

Spring hỗ trợ 2 cách cấu hình: XML và Java Base

Tùy chọn cấu hình dựa trên Java cho phép bạn viết hầu hết cấu hình Spring mà không cần XML

Sau đây là ví dụ 1 số annotation:

1. **@Configuration & @Bean**

* Định nghĩa @Configuration ở đầu class có nghĩa là class đó được quản lý bởi Spring IoC container như là 1 nơi để tạo ra Bean
* Định nghĩa @Bean ở đầu method có nghĩa là method đó return ra 1 object là Bean

VD:

@Configuration

public class HelloWorldConfig {

@Bean

public HelloWorld helloWorld(){

return new HelloWorld();

}

}

Đoạn code trên tương đương với nội dung XML sau

<beans>

<bean id = "helloWorld" class = "com.tutorialspoint.HelloWorld" />

</beans>

* Tên method cũng chính là Bean id, theo ví dụ trên thì Bean id là “helloWorld”
* Có thể định nghĩa nhiều Bean trong class
* Gọi và dùng các Bean bằng cách *AnnotationConfigApplicationContext*

public static void main(String[] args) {

ApplicationContext ctx = new AnnotationConfigApplicationContext(HelloWorldConfig.class);

HelloWorld helloWorld = ctx.getBean(HelloWorld.class);

helloWorld.setMessage("Hello World!");

helloWorld.getMessage();

}

1. **@Import**

Cho phép load định nghĩa Bean từ class config khác

**ConfigA.java**

@Configuration

public class ConfigA {

@Bean

public A a() {

return new A();

}

}

**ConfigB.java**

@Configuration

@Import(ConfigA.class)

public class ConfigB {

@Bean

public B b() {

return new B();

}

}

public static void main(String[] args) {

ApplicationContext ctx = new AnnotationConfigApplicationContext(ConfigB.class);

// now both beans A and B will be available...

A a = ctx.getBean(A.class);

B b = ctx.getBean(B.class);

}