**Giới thiệu**

Khi xây dựng một chương trình với **Spring Boot** đôi lúc chúng ta một **Bean** chỉ được load lên hoặc khởi tạo theo một điều kiện nào đó. Ví dụ như tạo một **Bean** trong môi trường Test, còn môi trường thật sẽ không cần nữa.

**Spring Boot** hỗ trợ chúng ta làm điều này với Annotation @Conditional.

Bài này yêu cầu kiến thức cơ bản:

1. [「Spring Boot #0」 Series làm chủ Spring Boot, Zero to Hero](https://loda.me/spring-boot-0-series-lam-chu-spring-boot-zero-to-hero-loda1558963914472)

Toàn bộ ví dụ trong bài viết này [đều có tại Github](https://github.com/loda-kun/spring-boot-learning)

**Cách tạo bean có điều kiện**

Trong **Spring Boot**, có rất nhiều cách để tạo ra Bean, bạn biết cách nào?

@Component, @Configuration, @Bean, @Service, v.v...

Với tất cả các cách tạo ra Bean, bạn đều có thể thêm **một hoặc nhiều điều kiện** để cho việc tạo ra Bean đó chỉ xảy ra nếu nó thỏa mãn một điều kiện cho trước.

**SpringBoot** hỗ trợ rất nhiều loại điều kiện khác nhau, tất cả đều sẽ bắt đầu bằng tiền tố @Conditional.... Còn hậu tố thì chúng ta sẽ đề cập sau.

Cách thức hoạt động của tất cả @Condition như sau:

@Configuration

public class ConditionalOnBeanExample {

/\*

ABeanWithCondition chỉ được tạo ra khi @Condition thỏa mãn

\*/

@Bean

@Conditional...

ABeanWithCondition aBeanWithACondition() {

return new ABeanWithCondition();

}

}

Nếu bạn gắn nó trên một @Configuration thì toàn bộ các @Bean bên trong sẽ bị chịu tác động.

@Conditional...

@Configuration

public class ConditionalOnBeanExample {

/\*

SomeOtherBean chỉ được tạo ra khi @Condition thỏa mãn

\*/

@Bean

SomeOtherBean someOtherBean() {

return new SomeOtherBean();

}

/\*

SomeOtherBean2 chỉ được tạo ra khi @Condition thỏa mãn

\*/

@Bean

SomeOtherBean2 someOtherBean2() {

return new SomeOtherBean2();

}

}

Tương tự cho tất cả các annotation khác như @Component, @Service, @Repository

@Conditional...

@Component

public class ConditionalOnBeanExample {

}

Bây giờ chúng ta sẽ đi tìm hiểu các loại điều kiện trong **Spring Boot**.

**@ConditionalOnBean**

@ConditionalOnBean sử dụng khi chúng ta muốn tạo ra một Bean, chỉ khi có một Bean khác đang tồn tại

@Configuration

public class ConditionalOnBeanExample {

/\*

Đây là một Bean, bạn hãy chạy ứng dụng

khi comment và chạy lại lần nữa nhưng bỏ dấu comment phía

dưới để xem kết quả.

\*/

// @Bean

RandomBean randomBean() {

return new RandomBean();

}

/\*

ABeanWithCondition chỉ được tạo ra, khi RandomBean tồn tại trong Context.

\*/

@Bean

@ConditionalOnBean(RandomBean.class)

ABeanWithCondition aBeanWithACondition() {

return new ABeanWithCondition();

}

}

**@ConditionalOnProperty**

Dùng @ConditionalOnProperty khi bạn muốn quyết định sự tồn tại Bean thông qua cấu hình property.

@Configuration

public class ConditionalOnPropertyExample {

/\*

@ConditionalOnProperty giúp gắn điều kiện cho @Bean dựa theo

property trong config

\*/

@Bean

@ConditionalOnProperty(

value="loda.bean2.enabled",

havingValue = "true", // Nếu giá trị loda.bean2.enabled = true thì Bean mới được khởi tạo

matchIfMissing = false) // matchIFMissing là giá trị mặc định nếu không tìm thấy property loda.bean2.enabled

ABeanWithCondition2 aBeanWithCondition2(){

return new ABeanWithCondition2();

}

}

*application.properties*

#Thay đổi thành true để tạo bean2

loda.bean2.enabled=true

**@ConditionalOnExpression**

Trong trường hợp bạn muốn thỏa mãn nhiều điều kiện trong property, hãy sử dụng @ConditionalOnExpression

@Configuration

@ConditionalOnExpression(

"${loda.expression1.enabled:true} and ${loda.expression2.enabled:true}"

)

public class ConditionalOnExpressionExample {

}

**@ConditionalOnMissingBean**

Nếu trong Context chưa tồn tại bất kỳ Bean nào tương tự, thì @ConditionalOnMissingBean sẽ thỏa mãn điều kiện và tạo ra một Bean như thế.

@Configuration

public class ConditionalOnMissingBeanExample {

/\*\*

\* Nếu trong Context chưa tồn tại một SomeMissingBean nào

\* Thì Bean ở dưới đây mới được tạo

\* @return

\*/

@ConditionalOnMissingBean

DataSource someMissingBean(){

return new DataSource();

}

}

**@ConditionalOnResource**

@ConditionalOnResource thỏa mãn khi có một resources nào đó do bạn chỉ định tồn tại

/\*

Nếu Spring Boot không tìm thấy file application.properties thì class này không được tạo

\*/

@Configuration

@ConditionalOnResource(resources = "/application.properties")

public class ConditionalOnResourceExample {

}

**@ConditionalOnClass**

@ConditionalOnClass thỏa mãn khi trong classpath có tồn tại class mà bạn yêu cầu

@Configuration

@ConditionalOnClass(name = "loda.me.spring.LodaHandsome")

class ConditionalOnClassExample {

}

**@ConditionalOnMissingClass**

@ConditionalOnMissingClass ngược lại với @ConditionalOnClass. thỏa mãn khi trong classpath **không** tồn tại class mà bạn yêu cầu

@Configuration

@ConditionalOnMissingClass(name = "loda.me.spring.LodaHandsome")

class ConditionalOnMissingClassExample {

}

**@ConditionalOnJava**

@ConditionalOnJava thỏa mãn nếu môi trường chạy version Java đúng với điều kiện

@Configuration

@ConditionalOnJava(JavaVersion.EIGHT)

class ConditionalOnJavaExample {

}

**Một số loại khác**

Vẫn còn các bạn ạ, những mình thấy nó ít khi được sử dụng, nên sẽ không đề cập, tránh loạn thông tin.

**Custom Conditional**

Tất nhiên là bạn hoàn toàn có thể tự tạo ra cho mình một điều kiện. Mình sẽ đề cập cách làm ở bài sau.

**Kết**