@Conditional là annotation khai báo ở đầu mỗi **bean** cho phép quyết định 1 bean có được tạo hay không phụ thuộc vào các điều kiện DEV xác định

- @ConditionalOnBean(value = {X.class,…class}) Nếu **value**  là các class đã được khởi tại trong context thì bean này sẽ được tạo

- @ConditionalOnJAVA(JAVAVERSION.EIGHT) Nếu **version Java hiện tại là 8** thì bean này sẽ được tạo

-@ConditionalOnMissingBean(bean X) Nếu **bean X** chưa được tạo thì **bean này** sẽ được khởi tạo

-@ConditionalOnMissingClass(Class X) Nếu **Class X** chưa được tạo thì **bean này** sẽ được khởi tạo  
- @ConditionalOnProperty(value)): nếu giá trị **value = true tại file config: \*.property / \*.yml**  thì Bean này sẽ được tạo

Vd: + @ConditionalOnProperty(value = "period",  
 havingValue = "12 years",  
 matchIfMissing = true)

=> matchIfMissing = true: Nếu không tồn tại tt period thì bean vẫn được tạo

+ @ConditionalOnProperty(value = "server.port",  
 havingValue = "8080",  
 matchIfMissing = false  
=> Nếu không tồn tại tt server.port thì bean không được tạo

),

@ConditionalOnExpression(“${value1} and/or ${value2}”) nếu cả 2 value đều là true, thbì bean này sẽ được khởi tạo.

Vd: @ConditionalOnExpression("${quanaqt.isSent:true} and ${quanzip.enable:true}")

* **Tự tạo @Conditional**

+ Tạo class X implemetn interface: **Condition,**

**Trên 1 bean bất kỳ: @Conditional(X.class,. class)**

**Vd:**

public class ConditionalOnSystem implements Condition {  
 @Override  
 public boolean matches(ConditionContext context, AnnotatedTypeMetadata metadata) {  
 System.*out*.println("OS information: " + System.*getProperty*("os.name"));  
 return System.*getProperty*("os.name").toLowerCase().contains("win");  
 }  
}

**sau đó với các bean:**

@Bean  
@Conditional(ConditionalOnSystem.class)  
// create this bean when app is running with java version 8  
@ConditionalOnJava(JavaVersion.*EIGHT*)  
// create this bean when quanaqt.isSent = true và quanzip.enable = true  
// Every thing in {} must be compare to true: dont use: ${quanaqt.isSent:FALSE} and ${quanzip.enable:true}  
@ConditionalOnExpression("${quanaqt.isSent:true} and ${quanzip.enable:true}")  
public QuanaqtConfig getAqtConfig() {  
 return new QuanaqtConfig();  
}

+Hoặc Tạo 1 @Annotation Y bên trên có các: **@Conditional(X.class,…class), @ConditionalOnJava,…Và trên các class cần điều kiện, chỉ cần đặt @Y**

**Vd :** import java.lang.annotation.\*;  
  
@Target({ElementType.*TYPE*, ElementType.*METHOD*})  
@Retention(RetentionPolicy.*RUNTIME*)  
@Documented  
  
@ConditionalOnJava(JavaVersion.*EIGHT*)  
@ConditionalOnMissingClass  
@Conditional({ConditionalOnSystem.class, ConditionalOnValue.class})  
public @interface ConditionOnEnv {  
}

**Sau đó với các bean:**

@Bean  
// Custom Conditional  
@ConditionOnEnv  
// create this bean when QuanzipConfig did not get created yet  
@ConditionalOnMissingBean  
// create this class when QuanzipConfig did not get created yet  
@ConditionalOnMissingClass  
public QuanzipConfig getZipConfig() {  
 return new QuanzipConfig();  
}