目标是用 **Cucumber + Spring Boot** 做测试，希望：

1. **支持依赖注入（@Autowired 等 Spring 特性）**
2. **根据不同环境读取配置文件（如 application-dev.yml, application-prod.yml 等）**

这完全可以实现，下面是推荐的配置方式：

**一、基本项目结构：**

bash

复制编辑

src

├── main

│ └── java/com/example/demo # 你的主应用代码

├── test

│ ├── java/com/example/demo

│ │ ├── config # Cucumber 配置

│ │ ├── steps # Step Definitions 实现

│ │ └── CucumberTest.java # Runner

│ └── resources

│ └── features # .feature 文件

**二、配置文件结构（多环境）**

yaml

复制编辑

# application.yml

spring:

profiles:

active: dev

yaml

复制编辑

# application-dev.yml

my:

config: "这是开发环境"

# application-prod.yml

my:

config: "这是生产环境"

**三、Cucumber 配合 Spring Boot 使用方法**

**1. 创建配置类 CucumberSpringConfiguration.java**

java

复制编辑

package com.example.demo.config;

import io.cucumber.spring.CucumberContextConfiguration;

import org.springframework.boot.test.context.SpringBootTest;

@CucumberContextConfiguration

@SpringBootTest

public class CucumberSpringConfiguration {

}

这个类是关键，让 Cucumber 能识别并使用 Spring 上下文。

**2. Step 文件中支持 @Autowired**

java

复制编辑

package com.example.demo.steps;

import com.example.demo.service.MyService;

import io.cucumber.java.en.Given;

import org.springframework.beans.factory.annotation.Autowired;

public class MyStepDefs {

@Autowired

private MyService myService;

@Given("打印当前配置")

public void printCurrentConfig() {

System.out.println("当前配置值: " + myService.getConfigValue());

}

}

**3. 使用 @Value 或 @ConfigurationProperties 读取配置：**

java

复制编辑

@Service

public class MyService {

@Value("${my.config}")

private String configValue;

public String getConfigValue() {

return configValue;

}

}

**4. Runner 类 CucumberTest.java**

java

复制编辑

package com.example.demo;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

import org.junit.runner.RunWith;

@RunWith(Cucumber.class)

@CucumberOptions(

features = "src/test/resources/features",

glue = {"com.example.demo.steps", "com.example.demo.config"},

plugin = {"pretty"}

)

public class CucumberTest {

}

**四、切换环境的方法**

方式有两种：

**方法1：在 application.yml 里设置默认激活的环境：**

yaml

复制编辑

spring:

profiles:

active: dev

**方法2：在命令行指定环境（推荐 CI/CD 时用）：**

bash

复制编辑

mvn test -Dspring.profiles.active=prod

**五、总结**

| **功能** | **配置方式** |
| --- | --- |
| 启用 Spring 注入 | @SpringBootTest + @CucumberContextConfiguration |
| 不同环境读取配置 | 使用 application-{profile}.yml |
| 注入配置值到类 | 使用 @Value 或 @ConfigurationProperties |
| 命令行切换环境 | -Dspring.profiles.active=xxx |

如果你愿意，我可以给你一个完整的最小可运行 demo（Spring Boot + Cucumber + 多环境配置）项目结构模板。你要用 Maven 还是 Gradle？我可以按你的方式来写。