springboot监听器使用方式

1. 注解注入

启动类添加@ServletComponentScan注解，监听器才能生效

**@SpringBootApplication**(scanBasePackages={**"com.first.study.\*"**})  
**@MapperScan**(**"com.first.study.\*\*.dao"**)  
**@ServletComponentScan**//Listerner 监听器，Filter过滤器，自定义servlet 启动扫描并注入  
public class Application {  
 //默认启动器  
 public static void main(String[] args) throws Exception {  
 SpringApplication.*run*(Application.class, args);  
 }}

import javax.servlet.ServletContextEvent;  
import javax.servlet.ServletContextListener;  
import javax.servlet.annotation.**WebListener**;**@WebListener**public class TestListerner implements ServletContextListener{  
 **@Override** public void contextInitialized(ServletContextEvent sce) {  
 System.*err*.println(**"初始化监听器。。。"**);  
 }  
 **@Override** public void contextDestroyed(ServletContextEvent sce) {  
 System.*err*.println(**"销毁监听器。。。"**);  
 }  
}

1. 手动注入

TestListerner不需要@WebListener注解

Application不需要@ServletComponentScan注解

import org.springframework.boot.web.servlet.ServletListenerRegistrationBean;  
import org.springframework.context.annotation.**Bean**;  
import org.springframework.stereotype.**Component**;  
import javax.servlet.ServletContextListener;  
**@Component**public class TestListernerConfig {  
 **@Bean** public ServletListenerRegistrationBean<ServletContextListener> servletListenerRegistrationBean(){  
 ServletListenerRegistrationBean<ServletContextListener> registrationBean=new ServletListenerRegistrationBean<>();  
 TestListerner testListerner=new TestListerner();  
 registrationBean.setListener(testListerner);  
 return registrationBean;  
 }  
}

**3注解方式，推荐使用此方式，**

Application不需要@ServletComponentScan注解

import org.springframework.stereotype.**Component**;  
import javax.servlet.ServletContextEvent;  
import javax.servlet.ServletContextListener;  
import javax.servlet.annotation.**WebListener**;**@WebListener  
@Component**//添加进IOC容器  
public class TestListerner implements ServletContextListener{  
 **@Override** public void contextInitialized(ServletContextEvent sce) {  
 System.*err*.println(**"初始化监听器。。。"**);  
 }  
 **@Override** public void contextDestroyed(ServletContextEvent sce) {  
 System.*err*.println(**"销毁监听器。。。"**);  
 }  
}

综上三种其实就是怎么将监听类注入到IOC容器中