В Spring Framework, аспекты представляют собой модули, содержащие перехватчики (advice) и точки среза (pointcuts), позволяющие внедрять поведение в приложение. Аспекты часто используются для решения поперечных задач, таких как логирование, транзакции, безопасность и другие аспекты, которые могут быть применены ко многим компонентам.

Пример аспекта для логирования в вашем проекте может выглядеть следующим образом:

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

private Logger logger = LoggerFactory.getLogger(this.getClass());

@Before("execution(\* de.example.bankexchange.service.\*.\*(..))")

public void logBefore(JoinPoint joinPoint) {

logger.info("Method {}() is about to be executed.", joinPoint.getSignature().getName());

}

@After("execution(\* de.example.bankexchange.service.\*.\*(..))")

public void logAfter(JoinPoint joinPoint) {

logger.info("Method {}() has been executed.", joinPoint.getSignature().getName());

}

}

В этом примере аспект LoggingAspect будет применяться ко всем методам в пакете **de.example.bankexchange.service** и логировать информацию до и после их выполнения.