

## Purpose of LoggingAspect Class ✓

The class is designed to intercept and log method calls in the service layer. This allows you to centralize logging logic instead of writing it in every service method.

## Key Components & Explanations

### 1. @Aspect

- What it does: Marks this class as an aspect — a modularization of cross-cutting concerns (like logging).
- Why it's important: Spring knows to scan and process this class using AspectJ.

### 2. @Component

- What it does: Registers this class as a Spring Bean so Spring can manage its lifecycle.
- Why it's important: Required for @Aspect to be detected and woven into Spring's AOP proxy system.

### 3. `Logger logger = LoggerFactory.getLogger(...)`

- What it does: Provides a thread-safe logger using SLF4J, the standard logging facade.
- Why it's important: It allows clean and formatted logging throughout your application.

## AOP Advices Used

### 4. @Before(...)

Purpose: Executes before the matched method.

Expression:

```
@Before("execution(* com.expertise.aop.example.service.*(..))")
```

- What it Matches: Any method in any class inside the service package, with any return type and parameters.
- Why it's Important: Useful to log method entry points or for performing pre-check logic.

### 5. @After(...)

Purpose: Executes after the matched method completes — whether successfully or with an exception.

Expression:

```
@After("execution(* com.expertise.aop.example.service.*(..))")
```

- Why it's Important: Useful for logging method completion or handling cleanup operations.

### 6. @AfterReturning(...)

Purpose: Executes only after the matched method successfully returns a result.

Expression:

```
@AfterReturning(pointcut = "execution(* com.expertise.aop.example.service.*(..))", returning = "result")
```

- Explanation: The "returning = result" part binds the returned object to the result parameter.
- You can use this result for logging, auditing, or further processing.
- Why it's Important: Helps trace return values for monitoring, debugging, or analytics.