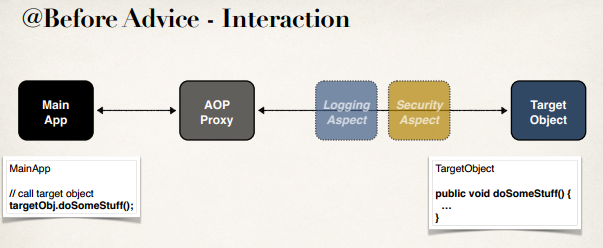
**AOP @Before Advice Type**

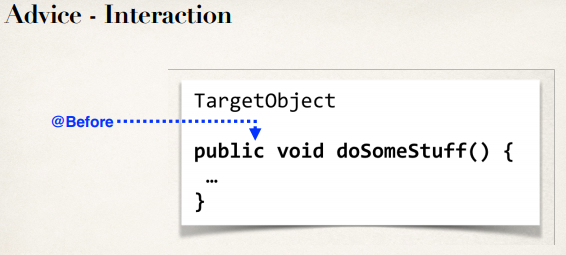
**Advice types**:

* **Before advice**: run before the method
* **After returning advice**: run after the method (success execution)
* **After throwing advice**: run after method (if exception thrown)
* **After finally advice**: run after the method (finally)
* **Around advice**: run before and after method

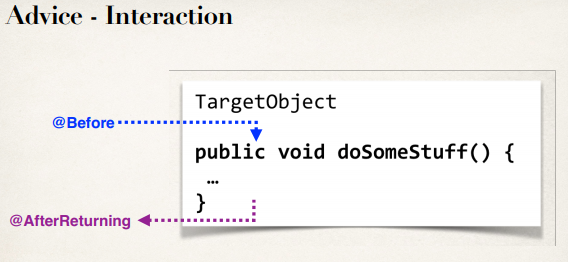
**@Before Advice – Interaction**:



**Advice – Interaction**:



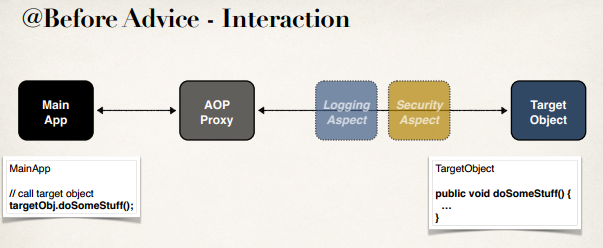
**Advice – Interaction – After Returning**:



**@Before Advice - Use Cases**:

* **Most common**
  + logging, security, transactions
* **Audit logging**
  + who, what, when, where?
* **API Management**
  + how many times has a method been called user?
  + analytics: what are peak times? what is average load? who is top user?

**AOP Example – Overview (Step-by-Step)**:



**Adding AspectJ JAR File**:

* Need to download AspectJ JAR file
* Even though we are using Spring AOP … still need AspectJ JAR file
* Why?
  + Spring AOP uses some of the AspectJ annotations
  + Spring AOP uses some of the AspectJ classes

**Development Process - @Before (Step-by-Step)**:

1. Create target object: AccountDAO
2. Create Spring Java Config class
3. Create main app
4. Create an Aspect with @Before advice

**Step 1: Create Target Object: AccountDAO**:

AOP @Before Advice Type