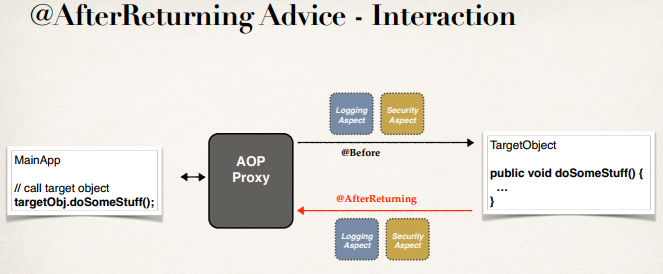
**41.1 AOP @AfterReturning Overview**

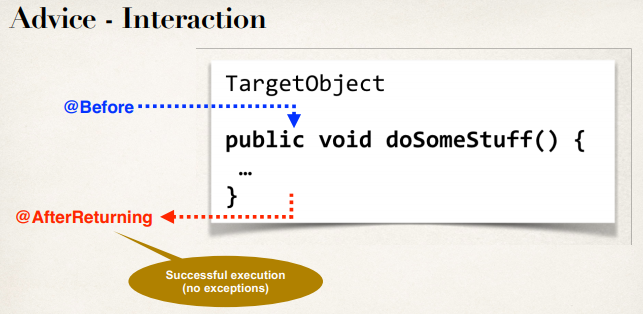
**Advice Types**:

1. **Before advice:** run before the method
2. **After returning advice:** run after the method (success execution)
3. **After throwing advice:** run after method (if exception thrown)
4. **After finally advice:** run after the method (finally)
5. **Around advice:** run before and after method

**@AfterReturning Advice – Interaction**:



**Advice – Interaction**:

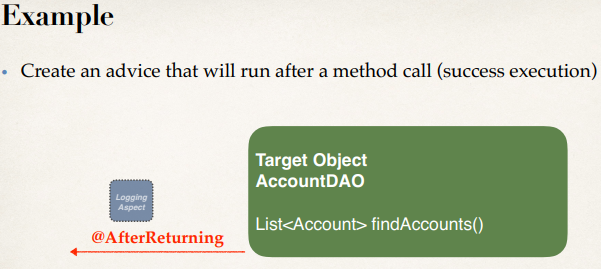


**AfterReturning Advice - Use Cases**:

1. **Most common**
   1. logging, security, transactions
2. **Audit logging**
   1. who, what, when, where?
3. **Post-processing Data**
   1. Post process the data before returning to caller
   2. Format the data or enrich the data (really cool but be careful)

**Example**:

Create an advice that will run after a method call (success execution)



**@AfterReturning Advice**:

This advice will run after the method call (success execution).

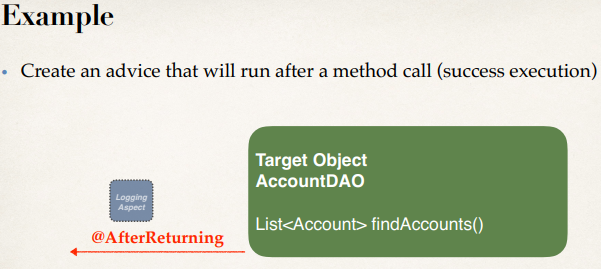
@AfterReturning(com.ruhul.aopdemo.dao.AccountDAO.findAccounts(..))")

**public** **void** afterReturningFindAccountAdvice(){

System.***out***.println("\n======>>> Executing @AfterReturning advice");

}

**Access the Return Value**:



**Access the Return Value**:

@AfterReturning(

pointcut = "execution(\* com.ruhul.aopdemo.dao.AccountDAO.findAccounts(..))", returning = "result")

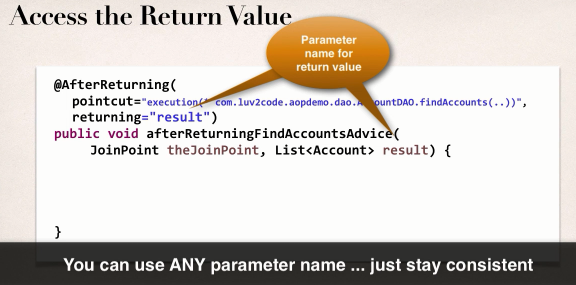
**public** **void** afterReturningFindAccountAdvice(

JoinPoint theJoinPoint, List<Account> result) {

// print out the result of the method call

System.***out***.println("\n======>>> result is: " + result);

}



**Best Practices: Aspect and Advices**:

1. Keep the code small
2. Keep the code fast
3. Do not perform any expensive / slow operations
4. Get in and out as QUICKLY as possible

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

1. Prep Work: Add constructors to Account class
2. Add new method: **findAccounts()** in **AccountDAO**
3. Update main app to call the new method: **findAccounts()**
4. Add **@AfterReturning** advice

41.1 AOP @AfterReturning Overview