

@Around

- Define another method in the class `StudentAspect` tied to the `log()` point cut and `args(id)` of type `long`. For the advice to be triggered it must match both parts of this compound expression. The advice method is defined as below;

```
@Around("log() && args(id)")
public Object around(ProceedingJoinPoint jp, long id) throws Throwable {
    LOG.info("Around before->" + jp.getSignature().getName() + " with id " + id);
    Student student = (Student) jp.proceed();
    LOG.info("Around after->" + student.getFirstName() + " " + student.getSurname() );
    return student;
}
```

- Note the `ProceedingJoinPoint` to get information of the target method being invoked and the `proceed` method to delegate down from the proxy to the target bean. We also have see both the argument going in and the output going out.

Trigger the @Around

- Only one method in our service first the bill, Student get(long id). Relaunch the Spring Boot Application, select a student in the Angular app and view the console

StudentAspect	: Around before->get with id 6
StudentAspect	: Invoked Method Before->get
StudentAspect	: Invoked Method After->get
StudentAspect	: Around after->Paul Cruthers