

# Throwing Exceptions with Mocks



# Throwing Exceptions

- May need to configure the mock to throw an exception
- Possible use case
  - Testing how the code handles exceptions



# Refresher

MockAnnotationTest.java

```
...  
  
@SpringBootTest(classes=MvcTestingExampleApplication.class)  
public class MockAnnotationTest {  
  
    @MockBean  
    private ApplicationDao applicationDao;  
  
    @Autowired  
    private ApplicationService applicationService;  
  
    ...  
  
}
```

We will mock the DAO to throw exceptions

# Throw Exception

MockAnnotationTest.java

```
@DisplayName("Thrown an Exception")
@Test
public void throwAnException() {
    CollegeStudent nullStudent = (CollegeStudent) context.getBean("collegeStudent");

    when(applicationDao.checkNotNull(nullStudent))
        .thenReturn(new RuntimeException());

    assertThrows(RuntimeException.class, () -> {
        applicationService.checkNotNull(nullStudent);
    });
}
```

When the checkNull(...) method is called ...  
thenReturn an exception

Assert that the exception was thrown

# Throw Exception: Consecutive calls

MockAnnotationTest.java

```
@DisplayName("Multiple Stubbing")
@Test
public void stubbingConsecutiveCalls() {
    CollegeStudent nullStudent = (CollegeStudent) context.getBean("collegeStudent");

    when(applicationDao.checkNotNull(nullStudent))
        .thenReturn(new RuntimeException())
        .thenReturn("Do not throw exception second time");

    assertThrows(RuntimeException.class, () -> {
        applicationService.checkNotNull(nullStudent);
    });

    assertEquals("Do not throw exception second time", applicationService.checkNotNull(nullStudent));
}
```

First call

First call ... throw exception

Consecutive calls ... do NOT throw exception  
Just return a string

Second call