



Rich Dammkoehler & Justin Searls
with the assistance of others.

Quick Terminology

Test Double

Stub

- return what you want
- throw an exception
- invoke custom behavior
- match arguments conditionally

Mock

- verifies that expected invocations occur
- fail tests when touched in an unexpected way
- match arguments conditionally

Spy

- verify expected invocations occur
- quietly remembers all invocations
- match arguments conditionally

Others

- Fake
- Dummy
- Override
- Partial Mock
- Proxy

Meanwhile, in Mockito

- A “mock” is really a test spy that can stub
- A “spy” is really more like a partial mock
- Confused? No worries, that’s common!

Stub with “when”

```
public class AddressSplitterTest {  
  
    @InjectMocks private AddressSplitter subject = new AddressSplitter();  
  
    @Mock private AddressInputQueue addressInputQueue;  
  
    ...  
  
    @Test  
    public void splitsAddressesByComma() {  
        when(addressInputQueue.next()).thenReturn("jim@weirich.com,kent@beck.com");  
  
        List<String> result = subject.split();  
  
        assertThat(result, hasItems("jim@weirich.com", "kent@beck.com"));  
    }  
}
```



Stub!

<https://gist.github.com/864185>

Verify behavior

```
public class MailDelivererTest {  
  
    @InjectMocks private MailDeliverer subject = new MailDeliverer();  
  
    @Mock private ExternalMailSystem externalMailSystem;  
  
    ...  
  
    @Test  
    public void sendsEmail() {  
        String expectedBody = "Hi Tim!";  
  
        subject.deliver("tim@wingfield.com", expectedBody);  
  
        verify(externalMailSystem).send("wingfield.com", "tim", expectedBody);  
    }  
}
```

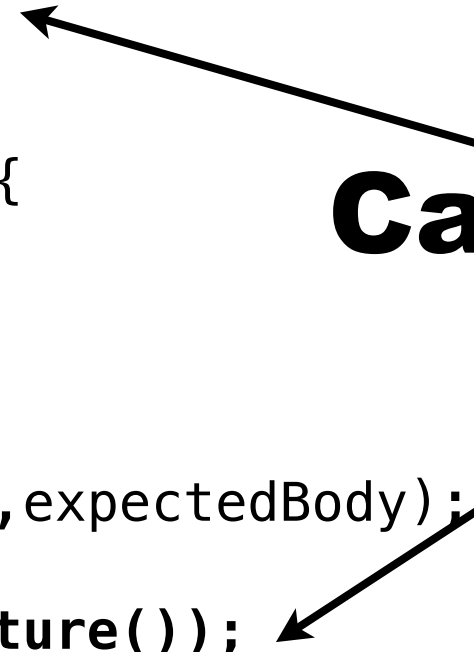
Verify!



<https://gist.github.com/864209>

Capture

```
public class MailDelivererTest {  
  
    @InjectMocks private MailDeliverer subject = new MailDeliverer();  
  
    @Mock private ExternalMailSystem externalMailSystem;  
  
    @Captor private ArgumentCaptor<Email> emailCaptor;  
    ...  
    @Test  
    public void sendsEmailByConstructingEmailObject() {  
        String expectedUser = "tim";  
        String expectedDomain = "wingfield.com";  
        String expectedBody = "Hi Tim!";  
  
        subject.deliver(expectedUser+"@"+expectedDomain,expectedBody);  
  
        verify(externalMailSystem).send(emailCaptor.capture());  
        Email email = emailCaptor.getValue();  
        assertThat(email.getUser(),is(expectedUser));  
        assertThat(email.getDomain(),is(expectedDomain));  
        assertThat(email.getBody(),is(expectedBody));  
    }  
}
```



Capture!

<https://gist.github.com/864224>

Pulling it all together

```
public class MailDelivererTest {  
  
    @InjectMocks private MailDeliverer subject = new MailDeliverer();  
  
    @Mock private ExternalMailSystem externalMailSystem;  
    @Mock private Timestampers timestampers;  
  
    @Captor private ArgumentCaptor<Email> emailCaptor;  
    ...  
    @Test  
    public void setsTheTimestampOnTheEmail() {  
        Date expected = new Date(89321);  
        when(timestampers.stamp()).thenReturn(expected);  
  
        subject.deliver("a@b", null);  
  
        verify(externalMailSystem).send(emailCaptor.capture());  
        Email email = emailCaptor.getValue();  
        assertThat(email.getTimestamp(), is(expected));  
    }  
}
```

Stub!

Capture!

Verify!

<https://gist.github.com/864276>

Mockito's Goals

- Only say what you want your test to specify
- Separate concepts of stubbing (“when”) and verification (“verify”)
- Preserve arrange-act-assert pattern
- Low-friction, sensible default behavior
- Descriptive errors / Clean stack traces

What's to like?

- The syntax (minimal, plain language, no replay)
- No mingling assertions into test setup
- The documentation!
- Annotations make it even easier than injecting real dependencies

Resources

- <http://mockito.org>
- <http://dannorth.net/2008/09/14/the-end-of-endotesting/>
- <http://xunitpatterns.com/Mocks,%20Fakes,%20Stubs%20and%20Dummies.html>
- <http://connextra.com/aboutUs/mockobjects.pdf?>
- <https://github.com/searls/mockito-testng-example>
(the sample project created for these slides)