

Converting Easymock-Tests to Mockito



↕

Your Rating: ☆☆☆☆☆ Results: ★★★★★ 25 rates

Converting Easymock-Tests to Mockito is pretty easier (much easier than vice versa).

Here's what we usually do:

Imports

- remove static imports to all Easymock & Easymock-Classextension classes
- add import static org.mockito.Mockito.*;

Creation

- replace all *create*Mock(<SomeClass>)* with *mock(<SomeClass>)*

Expectation

- replace all *expect()* with *when()*
- replace all *andReturn()* with *thenReturn()*

Verification

This is probably the point where you should care most. In EasyMock you set up verifications right between your expectations and verify them at the end of your test after mocks have been replayed.

EasyMockSnippet.java

```
MyClass mock = mock(MyClass.class);
// some expectations
mock.setTitle("Title");
// more expectations

classUnderTest.doSomethingThatInternallyCallsSetTitleOnMock(mock);
verify(mock);
```

With Mockito you add explicit verifications after the method to be tested has been called - which makes it much easier to use a BDD like GIVEN - WHEN - THEN approach

MockitoSnippet.java

```
MyClass mock = mock(MyClass.class);
when(mock.getName()).thenReturn("something");
..
..
classUnderTest.doSomethingThatInternallyCallsSetTitleOnMock(mock);
verify(mock).setTitle("Title");
```

So when converting you should do the following:

- identify all calls of void methods on your mocks (myMock.setSomething("value"))
- move these calls down to the // THEN section (after the call to the method to be tested)
- replace the "myMock"s with verify(myMock)

Caution: Mockito's mock() creates "nice" mocks per default - in case you really want to use strict mock's you'll have to do additional steps.

For additional info see [the Mockito-Wiki](#).