



Mock-framework

Team

Vasilev Pavel

Patrushev Borya

Goal

Mock-framework for Java:

- Creating mock objects using `mock()` and `spy()` (with appropriate annotations)
- Stubbing: semantically the same as *`when(myObject.method(param)).thenReturn(value)`* in Mockito (+ argument matching + *`thenThrow`* etc.)
- Call verification with *`verify + times..`*
- Mocking private/static/final methods, constructors, final classes, dynamic methods, interfaces



Alternatives

- jMock
- EasyMock
- PowerMock
- Mockito

☆	Star	126	▼
☆	Star	760	▼
☆	Star	3.8k	▼
☆	Star	12.7k	▼

BSD

THE APACHE SOFTWARE FOUNDATION

THE APACHE SOFTWARE FOUNDATION

MIT

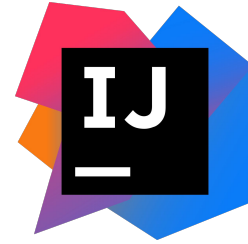
Static Static

🔒



Tools

- Java
- ByteBuddy
- Objenesis
- IntelliJ IDEA
- GitHub



Creating mock objects

```
//Mock creation with mock method  
Test test = Tarjun.mock(Test.class);
```

```
//Mock creation with annotation  
@Mock  
Test test;
```



Creating spy objects

```
//Spy creation with spy method  
List<String> testList = new ArrayList<>();  
List<String> testListSpied = Tarjun.spy(testList);
```

```
//Spy creation with spy annotation  
@Spy  
List<String> testListSpied = new ArrayList<>();
```



Method stubbing

```
when(test.foo()).thenReturn( retObj: "Tarjun");  
Assertions.assertEquals( expected: "Tarjun", test.foo());
```

```
@Override  
public String foo() { return "Foo"; }
```

```
testListSpied.add("Mockito");  
when(testListSpied.get(0)).thenReturn( retObj: "Tarjun");  
  
Assertions.assertEquals( expected: "Tarjun", testListSpied.get(0));
```



Argument matching

```
when(test.multi(anyInt(), eq(t: 10))).thenReturn(retObj: 42);  
when(test.multi(eq(t: 10), anyInt())).thenReturn(retObj: 1337);  
when(test.multi(a: 10, b: 10)).thenReturn(retObj: 7);  
  
Assertions.assertEquals(expected: 7, test.multi(a: 10, b: 10));  
Assertions.assertEquals(expected: 42, test.multi(a: 1, b: 10));  
Assertions.assertEquals(expected: 1337, test.multi(a: 10, b: 1));  
Assertions.assertEquals(expected: 0, test.multi(a: 1, b: 1));
```

```
public int multi(int a, int b) { return a * b; }
```



thenThrow

```
when(test.multi( a: 1, b: 2)).thenThrow(new IllegalArgumentException());
```

invokeRealMethod

```
when(testListSpied.contains("Mock-framework")).thenReturn(Boolean.TRUE);  
when(testListSpied.contains("Kock")).invokeRealMethod();  
Assertions.assertFalse(testListSpied.contains("Kock"));  
Assertions.assertTrue(testListSpied.contains("Mock-framework"));
```

Segregation of duties

- Borya - mock&spy objects with annotations, stubbing, thenThrow, tests
- Pavel - argumentMatchers, static/private methods, tests



Demo

Thank you for your attention

