



5 Inspecting Calls

Inspecting Calls

Mock records all calls made along with arguments. Assertion methods can make claims about how calls are made. However, they're unable to provide detailed insights into calls.

Mocks include attributes used to access more detailed call information.

The `unittest.mock` module includes a helper object named `call` used as a means of creating calls for comparison against recorded calls. Providing `call` the same arguments provided to a **mock call** makes the two calls comparable.

The below code demonstrates how to interact with recorded calls.

```
1 from unittest.mock import call, Mock
2 ##### Inspecting Calls #####
3 #
4 fake_object = Mock()
5 #
6 # Call with positional and keyword arguments.
7 fake_object('hi', 'there', who='world')
8 # Call without arguments.
9 fake_object()
10 # The call_count attribute reflects the number of calls made to
11 assert fake_object.call_count == 2
12 # Inspecting calls more closely is done with the unittest.mock.
13 # Call makes it easier to make assertions regarding how the mock
14 #
15 # Comparisons can be made by providing the same arguments to call
16 # provided to the mock.
17 #
18 # The calls to fake_object produce the following list of calls.
19 expected_calls = [call('hi', 'there', who='world'), call()]
20 # The assert_has_calls ensures that calls made to mock match what
21 # By default the order of the calls must match the order they were
22 # made in.
```

< Back

Start check ↻



```
18 # The calls to fake_object produce the following list of call
19 expected_calls = [call('hi', 'there', who='world'), call()]
20 # The assert_has_calls ensures that calls made to mock match wh
21 # By default the order of the calls must match the order they w
22 fake_object.assert_has_calls(expected_calls)
23 # The any_order keyword argument ignores the call order.
24 fake_object.assert_has_calls(expected_calls, any_order=True)
25 # The call_args_list attribute stores the calls made to the moc
26 # order they were called.
27 assert fake_object.call_args_list == expected_calls
28 # Call the mock again so that this is the last call made.
29 fake_object('a', 'z', lang='en', case='lower')
30 # The call_args attribute stores the arguments of the last call
31 # The first element contains positional arguments.
32 # The second element contains keyword arguments.
33 assert fake_object.call_args == (('a', 'z'), {'lang': 'en', 'c
34 # Specific elements are accessible by name.
35 assert fake_object.call_args.args == ('a', 'z')
36 assert fake_object.call_args.kwargs == {'lang': 'en', 'case':
37 # Mock doesn't record method calls in the call_args_list attrib
38 # Calling a mock method.
39 fake_object.fake_attribute.fake_method("a")
40 # The method_calls attribute displays only method calls and not
41 # made directly to the mock object.
42 assert fake_object.method_calls == [call(fake_attribute.fake_me
43 # The mock_calls attribute displays all calls.
44 assert fake_object.mock_calls == [
45     call("hi", "there", who="world"),
46     call(),
47     call("a", "z", lang="en", case="lower"),
48     call(fake_attribute.fake_method("a"),
49 ]
50 #####
51 print('No assertion errors')
```

Instructions

< Back

Start check ↻

