

API Design A4

GitHub repo link: <https://github.com/aakshu/a4>

Testing code and screenshots

Mocked tests

Code:

Screenshot of successful test:

Live tests

Code:

Screenshot of successful test:

Testing code and screenshots

Mocked tests

Code:

Test:

```
# test for get_book_titles() with mocks
def test_mock_get_book_titles(self):
    # create a mock test client
    mock_test_client = MagicMock()
    # define behavior when get_book() function of mock client is called
    mock_test_client.get_book.side_effect = get_book_for_isbn

    # list of isbn to query
    list_of_isbn = ["1000", "1001"]
    # retrieve list of titles using mock client
    list_of_titles = get_book_titles(mock_test_client, list_of_isbn)

    # assert that the response is as expected
    self.assertEqual(list_of_titles, ["Harry Potter 3", "Percy Jackson"])
```

Helper function:

```
# helper function to return a mock book when mock client calls get_book()
def get_book_for_isbn(isbn):
    # create a mock book
    mock_book = MagicMock()

    # create mock book with title based on the isbn input
    if isbn == "1000":
        mock_book.title = "Harry Potter 3"
    elif isbn == "1001":
        mock_book.title = "Percy Jackson"
    elif isbn == "1002":
        mock_book.title = "Hunger Games"
    else:
        mock_book.title = "Invalid ISBN"

    # return the mock book
    return mock_book
```

Screenshot of successful test:

```
● aakshu@aakshusbookpro a4 % python client/test.py
.
-----
Ran 1 test in 0.001s

OK
○ aakshu@aakshusbookpro a4 %
```

Live tests

Code:

Test:

```
# test for live server method for get_book_titles()
# NOTE: this test will only work if the server is running
def test_live_get_book_titles(self):
    # create an instance of client class
    test_client = inventoryClient("localhost:50051")

    # list of isbn to query
    list_of_isbn = ["1000", "1001"]
    # retrieve list of titles using real client object that queries live server
    list_of_titles = get_book_titles(test_client, list_of_isbn)

    # assert that the response is as expected
    self.assertEqual(list_of_titles, ["Harry Potter 3", "Percy Jackson"])
```

Screenshot of successful test:

(running two tests because the first mocked test defined above is also run)

```
aakshu@aakshuscbookpro a4 % python client/test.py
..
-----
Ran 2 tests in 0.019s

OK
aakshu@aakshuscbookpro a4 %
```