

Completing the Testing Framework

Team Lucky 7

Laura Barber, Venessa Johansen-Barrera, Michael Blackburn

Introduction

Upon completion of Deliverable 3 our framework and script were functioning, but not as originally specified. After receiving in-class feedback we completely rewrote the script in order to have it in charge of all of the testing. We also simplified the driver so that it would only accept the method name and item being tested. Additionally, instead of having all of the tests in one file we broke them down into one text file for each test, each following the “Tested Items” format outlined below. After updating the original six test cases we added nineteen more to complete the required twenty-five test cases. We then ensured that the result of each test was compared to the defined oracle.

Otherwise it was a simple matter of some additional formatting changes being applied to the output results and there further organization of the framework itself.

Test Case Decisions

Coming up with the twenty-five test cases stated below was one of our easier decisions of this project. We knew that we were going to be testing the methods: nginx, django, and perl because those were the methods that were easily testable. So our first five test cases were based off of our familiarity of a URL and knowing what that particular URL was served under. To verify that our knowledge and our framework testing were correct outside of the methods, we used <https://www.whatismyip.com/server-headers-check/>. We generated the additional twenty test cases using the same methods, choosing several URLs for each, all with the same structure of using URLs we were familiar with and double checking our correctness with the above link. For further test case creation, we would suggest doing the same. Pick several URLs accustomed to or random URLs to test the methods along with using an outside source to double check.

Tested Items

The items will be tested and the results will be output in the format outlined below:

- **test number**
- **requirement tested**
- **method tested**
- **test inputs**
- **expected outcome**

Test Cases

Chapter 4

p2

1

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: https://pip.readthedocs.org/en/latest/

expected outcome: True

2

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://docs.fabfile.org/en/latest/

expected outcome: True

3

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://cs.cofc.edu

expected outcome: False

4

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://readthedocs.org/security/

expected outcome: True

5

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: http://www.microsoft.com

expected outcome: False

6

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://cs.cofc.edu

expected outcome: False

7

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://phpmyadmin.readthedocs.org/ja/latest/

expected outcome: True

8

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://phpmyadmin.readthedocs.org/cs/latest/

expected outcome: True

9

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://phpmyadmin.readthedocs.org/en/latest/

expected outcome: True

10

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: https://ericholschercom.readthedocs.org/en/latest/

expected outcome: True

11

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: https://ericholschercom.readthedocs.org/about/

expected outcome: True

12

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: https://ericholschercom.readthedocs.org/en/latest/about/

expected outcome: True

13

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://docs.pylonsproject.org/projects/pyramid/en/latest/

expected outcome: True

14

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://docs.pylonsproject.org/projects/pyramid-amon/en/latest/genindex.html

expected outcome: True

15

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://readthedocs.org/search/

expected outcome: True

16

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://readthedocs.org/api/v1/?format=json

Chapter 4

p4

expected outcome: True

17

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://readthedocs.org/accounts/login/

expected outcome: True

18

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: https://readthedocs.org/accounts/login/

expected outcome: True

19

requirement: url is served by perl and return 200 OK

method: **served_by_perl**(url)

test input: http://www.perl.org/

expected outcome: False

20

requirement: url is served by perl and return 200 OK

method: **served_by_perl**(url)

test input: https://ericholschercom.readthedocs.org/

expected outcome: False

21

requirement: url is served by perl and return 200 OK

method: **served_by_perl**(url)

test input: http://www.microsoft.com

expected outcome: False

22

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

test input: http://www.microsoft.com

expected outcome: False

23

requirement: url is served by perl and return 200 OK

method: **served_by_perl**(url)

test input: http://my.cofc.edu

expected outcome: False

24

requirement: url is served by nginx and return 200 OK

method: **served_by_nginx**(url)

Chapter 4

p5

test input: http://my.cofc.edu

expected outcome: False

25

requirement: url is served by django and return 200 OK

method: **served_by_django**(url)

test input: http://my.cofc.edu

True

Moving Forward

Further revisions will be made to the framework based upon feedback received in class, and these will be integrated prior to Deliverable 5. For the next deliverable, and given the design of the methods being tested for our project, there is little to change to break the code. Our primary changes will likely come from changing the HTTP status codes for valid urls from 200 to some other code, of which there are many options. Another possibility exists for breaking the code by changing the predefined lists of valid served urls to other, invalid urls. Both of these methods for breaking the code will allow individual cases to be broken, without breaking the entire framework.