

SOFTWARE TESTING METHODOLO HW9: TDI - Set up environment and application

Test Host: Digital Machine

Remote Address: <http://changcode.com>

Current Stack: Ubuntu + Nginx

pip wasn't be installed

```
shuchang@changcode:~$ git --version
git version 1.9.1
shuchang@changcode:~$ pip
The program 'pip' is currently not installed. You can install it by typing:
sudo apt-get install python-pip
shuchang@changcode:~$
```

Test plan:

1. Setup all python environment including: django, Selenium, firefox, etc.
 - 1.1 Install pip3
 - 1.2 Through pip3 install django
 - 1.3 Download todo project
 - 1.4 Install selenium
 - 1.5 Install firefox

I put all setup process into unit test class, running result and code like this:

```
Output
.....
-----
Ran 5 tests in 28.960s
OK
Start installing firefox
Start installing selenium
Start setting django djangoSpace
Start setting django
Start setting pip3
```

```

11
12 class Test_Setup(unittest.TestCase):
13
14     def test_setup_pip3(self):
15         print("Start setting pip3")
16         with cd("~"):
17             result = run("which pip3")
18             self.assertTrue("pip3" in result)
19
20
21     def test_setup_django(self):
22         print("Start setting django")
23         with cd("~"):
24             result = run("pip3 install django")
25         with cd("~"):
26             result = run("pip3 list | grep Django")
27
28         self.assertTrue(result.startswith("Django"))
29
30     def test_setup_application(self):
31         print("Start setting django djangoSpace")
32         with cd("~/djangoSpace"):
33             run("wget https://github.com/hjwp/book-example/archive/chapter_07.zip")
34             run("unzip chapter_07.zip")
35             run("mv book-example-chapter_07 application")
36         with cd("~/djangoSpace/application"):
37             run("python3 manage.py migrate")
38
39     def test_install_selenium(self):
40         print("Start installing selenium")
41         with cd("~"):
42             run("pip3 install selenium")
43         with cd("~"):
44             result = run("pip3 list | grep selenium")
45
46         self.assertTrue(result.startswith("selenium"))
47
48         with cd("~"):
49             run("apt-get install xvfb")
50             run("pip3 install pyvirtualdisplay")
51
52
53     def test_install_firefox(self):
54         print("Start installing firefox")
55         with cd("~"):
56             run("sudo apt-get update")
57             run("sudo apt-get install firefox")
58
59 if __name__ == "__main__":
60     unittest.main()

```

2. Start unit test in well configured remote environment:

2.1 test the user is myself:shuchang

2.2 verify python3 is installed

2.3 verify pip3 is installed

2.4 verify git is installed

2.5 verify django is installed

2.6 verify project is well configured

2.7 verify selenium is installed

2.8 verify firefox is installed

2.9 function test in project start

2.10 additional test python is running

```
14 def test_000_login(self):
15     "Test user name is correct"
16     with cd("~/djangoSpace"):
17         result = run("whoami")
18         self.assertTrue(env.user in result)
19
20 def test_001_python3(self):
21     "Python 3 is installed."
22     with cd("~"):
23         result = run(setup + "python3 --version")
24         self.assertTrue("Python 3" in result)
25
26 def test_002_pip3(self):
27     "Pip3 is installed."
28     with cd("~"):
29         result = run(setup + "pip3 --version")
30         self.assertTrue(result.startswith("pip"))
31
32 def test_003_git(self):
33     "git is installed."
34     with cd("~"):
35         result = run(setup + "git --version")
36         self.assertTrue(result.startswith("git"))
37
38 def test_004_django(self):
39     "Django is installed."
40     with cd("~"):
41         result = run(setup + "pip3 list | grep Django")
42         self.assertTrue(result.startswith("Django"))
43
44 def test_005_app_installed(self):
45     "Test that the application directory and software are present"
46     with cd("~/djangoSpace"):
47         result = run("ls -l")
48         self.assertTrue("application" in result)
49         with cd("~/djangoSpace/application"):
50             result = run("ls -l")
51             self.assertTrue("manage.py" in result)
52             self.assertTrue("lists" in result)
```

```
def test_006_selenium_installed(self):
    "Test Selenium webdriver is installed"
    with cd("~/djangoSpace"):
        result = run("ls -l")
    with cd("~/djangoSpace/application/functional_tests"):
        run("ls -l")
        result = run("python3 tests.py")
    print(result)

def test_007_firefox(self):
    "Firefox is installed."
    with cd("~"):
        result = run("firefox --v")
    self.assertTrue("Firefox" in result)

def test_008_function_test(self):
    "Function Test."
    with cd("~/djangoSpace/application/"):
        result = run("xvfb-run python3 manage.py test functional_tests")
    print(result)

def test_009_python(self):
    "Python"
    with cd("~/djangoSpace/"):
        result = run("xvfb-run python3 manage.py runserver")

def test_010_app_is_running(self):
    "Test that a Django application is running"
    with cd("~"):
        result = run("ps -aux | grep python")
    print(result)
    self.assertTrue("python" in result)
```

The final running result:

Output



```
test_000_login (__main__.Test_000_Environment)
Test user name is correct ... ok
test_001_python3 (__main__.Test_000_Environment)
Python 3 is installed. ... ok
test_002_pip3 (__main__.Test_000_Environment)
Pip3 is installed. ... ok
test_003_git (__main__.Test_000_Environment)
git is installed. ... ok
test_004_django (__main__.Test_000_Environment)
Django is installed. ... ok
test_005_app_installed (__main__.Test_000_Environment)
Test that the application directory and software are present ... ok
test_006_selenium_installed (__main__.Test_000_Environment)
Test Selenium webdriver is installed ... ok
test_007_firefox (__main__.Test_000_Environment)
Firefox is installed. ... ok
test_008_function_test (__main__.Test_000_Environment)
Function Test. ... ok
test_009_python (__main__.Test_000_Environment)
Python ... ok
test_010_app_is_running (__main__.Test_000_Environment)
Test that a Django application is running ... ok

-----

Ran 11 tests in 65.913s
OK
..
-----

Ran 2 tests in 37.833s
OK
Creating test database for alias 'default'...
Destroying test database for alias 'default'...
shuchang 28822  0.0  0.2  12396  1436 ?        Ss   02:08   0:00 bash -c cd ~ && ps
-aux | grep python
shuchang 28824  0.0  0.1  11740   916 ?        S    02:08   0:00 grep python
```

3. In the todo django project, the test file including:

There needs some configuration on web driver, set up display, then start testing.

```
6
7     def setUp(self):
8         self.display = Display(visible=0, size=(1024,768))
9         self.display.start()
10        self.browser = webdriver.Firefox()
11        self.browser.implicitly_wait(3)
12
13    def tearDown(self):
14        self.browser.quit()
15
16
17    def check_for_row_in_list_table(self, row_text):
18        table = self.browser.find_element_by_id('id_list_table')
19        rows = table.find_elements_by_tag_name('tr')
20        self.assertIn(row_text, [row.text for row in rows])
21
22
23    def test_can_start_a_list_and_retrieve_it_later(self):
24        # Edith has heard about a cool new online to-do app. She goes
25        # to check out its homepage
26        self.browser.get(self.live_server_url)
27
28        # She notices the page title and header mention to-do lists
29        self.assertIn('To-Do', self.browser.title)
30        header_text = self.browser.find_element_by_tag_name('h1').text
31        self.assertIn('To-Do', header_text)
32
33        # She is invited to enter a to-do item straight away
34        inputbox = self.browser.find_element_by_id('id_new_item')
35        self.assertEqual(
36            inputbox.get_attribute('placeholder'),
37            'Enter a to-do item'
38        )
39
40        # She goes to "Django Todo List" and enters a new to-do item (Edith's list)
```