How To Run Automation Test on Demo Todo Application

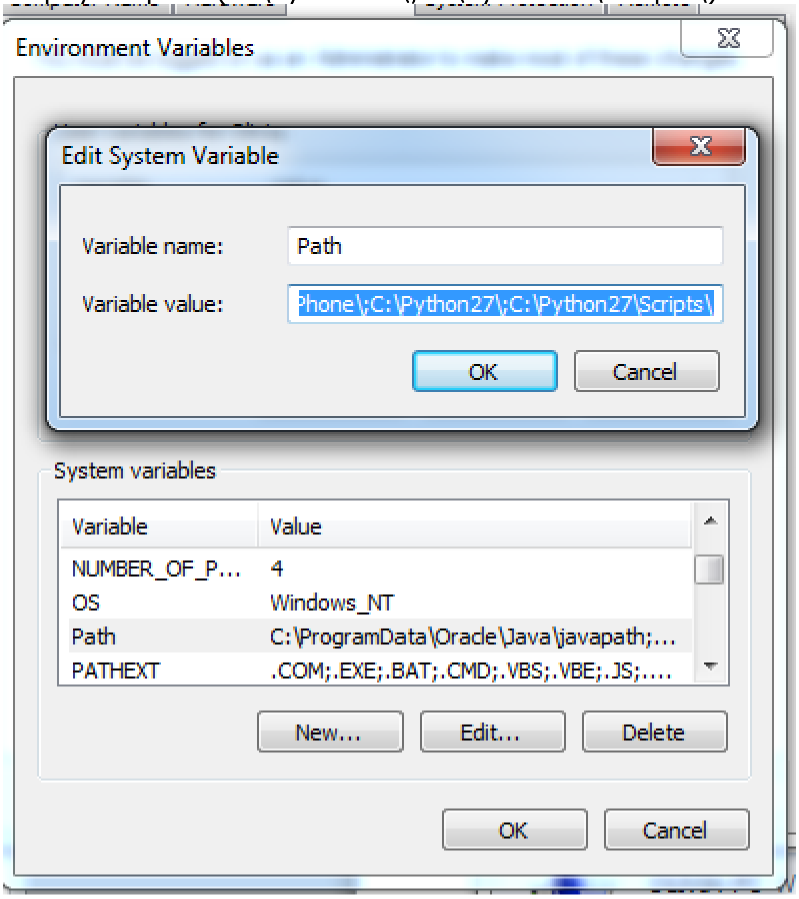
By: Olivia Meydina A.W

Current automation test are using Phyton and selenium.
Software needed to run this testing:

- Firefox
  Used as the browser for testing
- Notepad++
  Used to write the automation code
- Phyton 2.7.10

1. Installation and environment setup

- Download and install firefox and notepad++ by follow link above
- Download and install install phyton by follow link above
- Add phyton path (C:\Python27\;C:\Python27\Scripts\)in the environment variable



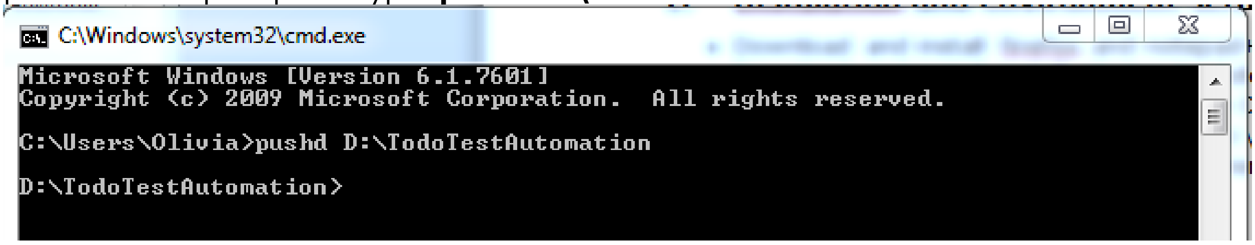
Variable name: Path

Variable value: Phone\;C:\Python27\;C:\Python27\Scripts\

Variable	Value
NUMBER_OF_P...	4
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

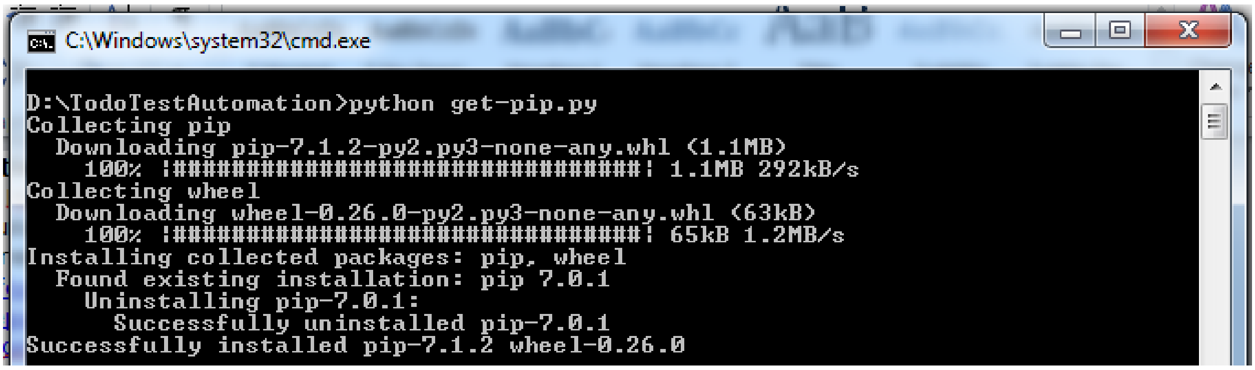
- After install the phyton and set the environment variable, open this link https://bootstrap.pypa.io/get-pip.py, copy all value, paste in notepad++ and save as get-pip.py. Copy get-pip.py file in D:\TodoTestAutomation

- Open command promp and type "**pushd D:\TodoTestAutomation**" on cmd



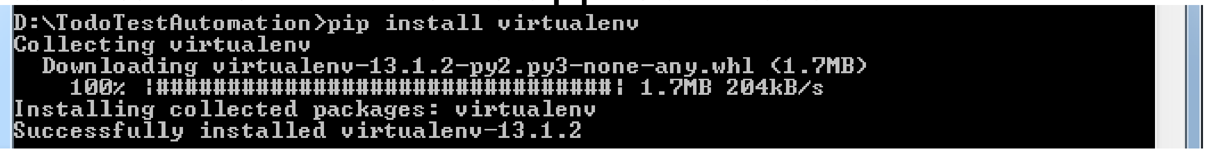
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Olivia>pushd D:\TodoTestAutomation

D:\TodoTestAutomation>
```

- Install pip by command **phyton get-pip.py** It will run the get-pip.py in TodoTestAutomation folder



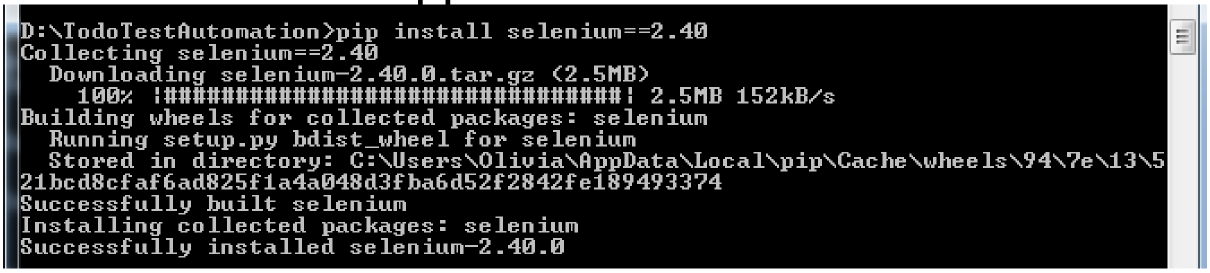
```
D:\TodoTestAutomation>python get-pip.py
Collecting pip
  Downloading pip-7.1.2-py2.py3-none-any.whl (1.1MB)
    100% |################################| 1.1MB 292kB/s
Collecting wheel
  Downloading wheel-0.26.0-py2.py3-none-any.whl (63kB)
    100% |################################| 65kB 1.2MB/s
Installing collected packages: pip, wheel
  Found existing installation: pip 7.0.1
    Uninstalling pip-7.0.1:
      Successfully uninstalled pip-7.0.1
Successfully installed pip-7.1.2 wheel-0.26.0
```

- Install virtual environment with command **pip install virtualenv**



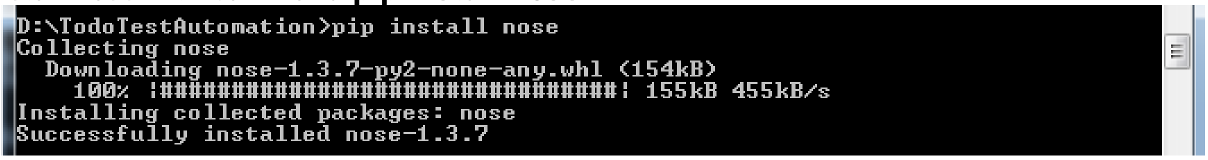
```
D:\TodoTestAutomation>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-13.1.2-py2.py3-none-any.whl (1.7MB)
    100% |################################| 1.7MB 204kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-13.1.2
```

- Install Selenium with command **pip install selenium==2.40**



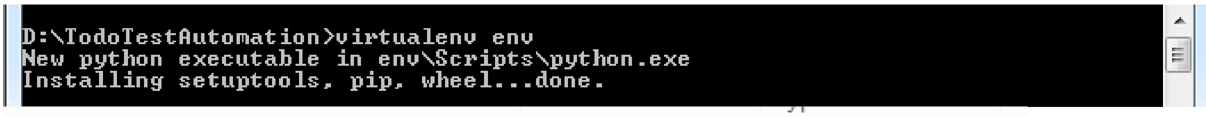
```
D:\TodoTestAutomation>pip install selenium==2.40
Collecting selenium==2.40
  Downloading selenium-2.40.0.tar.gz (2.5MB)
    100% |################################| 2.5MB 152kB/s
Building wheels for collected packages: selenium
  Running setup.py bdist_wheel for selenium
  Stored in directory: C:\Users\Olivia\AppData\Local\pip\Cache\wheels\94\7e\13\5
21bcd8cfaf6ad825f1a4a048d3fba6d52f2842fe189493374
Successfully built selenium
Installing collected packages: selenium
Successfully installed selenium-2.40.0
```

- Install nose with command **pip install nose**

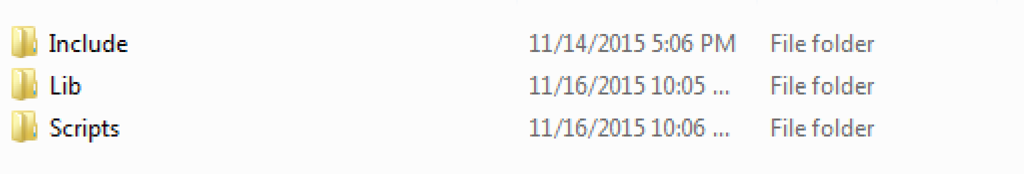


```
D:\TodoTestAutomation>pip install nose
Collecting nose
  Downloading nose-1.3.7-py2-none-any.whl (154kB)
    100% |################################| 155kB 455kB/s
Installing collected packages: nose
Successfully installed nose-1.3.7
```

- Create virtual environment with command **virtualenv env.** There will be a folder env under TodoAutomation with folders inside

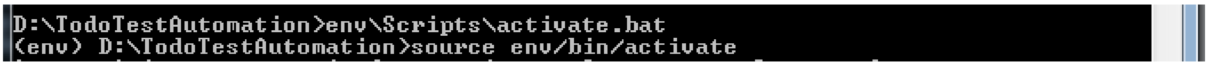


```
D:\TodoTestAutomation>virtualenv env
New python executable in env\Scripts\python.exe
Installing setuptools, pip, wheel...done.
```



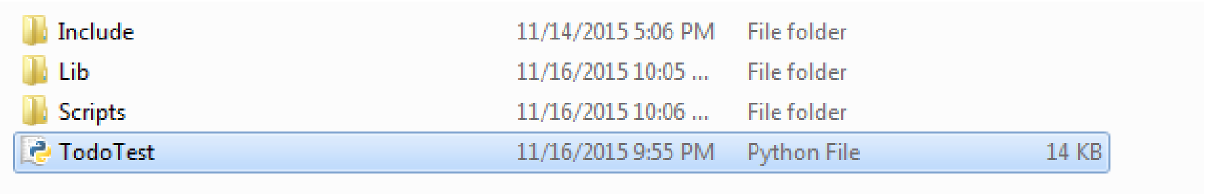
Include	11/14/2015 5:06 PM	File folder
Lib	11/16/2015 10:05 ...	File folder
Scripts	11/16/2015 10:06 ...	File folder

- Activate virtual environment



```
D:\TodoTestAutomation>env\Scripts\activate.bat
(env) D:\TodoTestAutomation>source env/bin/activate
```

- Copy **TodoTest.py** to D:\TodoTestAutomation\env\



Include	11/14/2015 5:06 PM	File folder	
Lib	11/16/2015 10:05 ...	File folder	
Scripts	11/16/2015 10:06 ...	File folder	
TodoTest	11/16/2015 9:55 PM	Python File	14 KB

After set up the environment you can continue to the test.

2. Testing and Test Verification

2.1 Test Cases

Here is the test cases used to test the website.

No	Test Idea	Test Step		Expected Result	Automation Test Name
		No	Step		
1	Check Add functionality	1	Open web browser	Web browser opened	test_add
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “List todo” and press enter key	“List Todo” listed in the list and information todo list updated “1 item left”	
2	Check Delete functionality	1	Open firefox	Firefox opened	test_delete
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “List delete” and press enter key	Data inputted to the list	
		5	Type “List delete 2” and press enter key		
		6	Go to fist item “List delete” and click delete button beside it.	Item deleted from the list, only 1 item listed and information todo list updated “1 item left”	
3	Check edit functionality	1	Open firefox	Firefox opened	test_edit
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “List Todo” and press enter key	Data inputted to the list	
		5	Double click inputted list and edit the value to “List Todo Edited”	Item updated to “List Todo Edited”	
4	Verify Uncheck an item functionality	1	Open web browser	Web browser opened	test_uncheckTodo
		2	Go to web page	Web page opened contains	

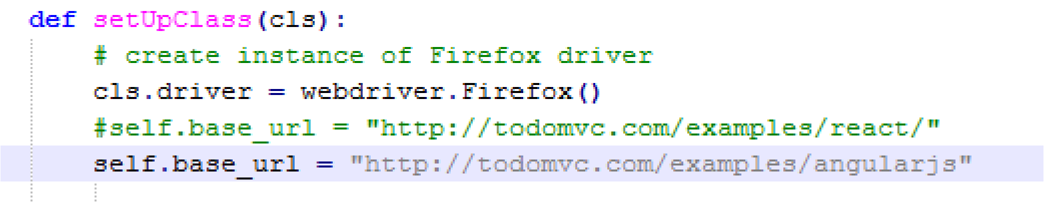
			http://todomvc.com/examples/react	‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “List Todo” and press enter key	Data inputted to the list	
		5	Check the checkbox beside the item	Item checked	
		6	Uncheck the checkbox beside the item	Item unchecked and information todo list updated “1 item left”	
5	Verify Check an item functionality	1	Open web browser	Web browser opened	test_checkTodo
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “List Todo” and press enter key	Data inputted to the list	
		5	Type “List Todo 2” and press enter key		
		6	Check the checkbox beside “List Todo 2”	Item checked and information todo list updated “1 item left”	
6	Verify Uncheck all functionality	1	Open web browser	Web browser opened	test_uncheckall
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	
		5	Type “Item Todo 2” and press enter key		
		6	Check using all checked beside “Whats need to be done?” text	All item checked	
			Uncheck using all checked beside “Whats need to be done?” text	All item unchecked and information todo list updated “2 items left”	
7	Verify check all functionality	1	Open web browser	Web browser opened	test_checkall
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	

		5	Type “Item Todo 2” and press enter key		
		6	Check using all checked beside “Whats need to be done?” text	All item checked	
8	Check “Active” button functionality	1	Open web browser	Web browser opened	test_active
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	
		5	Type “Item Todo 2” and press enter key		
		6	Type “Item Todo 3” and press enter key		
		7	Check the second item	Second item checked	
		8	Click “Active” button	The list updated. 2 item uncompleted displayed with no checkbox checked.	
9	Check “Completed” button functionality	1	Open web browser	Web browser opened	test_completed
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	
		5	Type “Item Todo 2” and press enter key		
		6	Type “Item Todo 3” and press enter key		
		7	Check the second item	Second item checked	
		8	Click “Completed” button	The list updated. All completed item shown (1 item) and the checkbox is checked	
10	Check “All” button functionality	1	Open web browser	Web browser opened	test_all
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	
		5	Type “Item Todo 2” and press enter key		

		6	Type “Item Todo 3” and press enter key		
		7	Check the second item	Second item checked	
		8	Press “All” button	All list displayed (3 items)	
11	Check “Clear completed” button functionality	1	Open web browser	Web browser opened	test_clearCompleted
		2	Go to web page http://todomvc.com/examples/react	Web page opened contains ‘todos’ as header	
		3	Click input box below ‘todos’ text	Input text got focus	
		4	Type “Item Todo” and press enter key	Data inputted to the list	
		5	Type “Item Todo 2” and press enter key		
		6	Type “Item Todo 3” and press enter key		
		7	Check the first and third item	2 items checked	
		8	Press “Clear completed” button	The list updated. All completed item is deleted, only 1 active item left with checkbox unchecked and information todo list updated “1 item left”	
12	Check testcase 1-11 in http://todomvc.com/examples/angularjs/		Do all test case but change step 2 web address to http://todomvc.com/examples/angularjs/	All step passed	

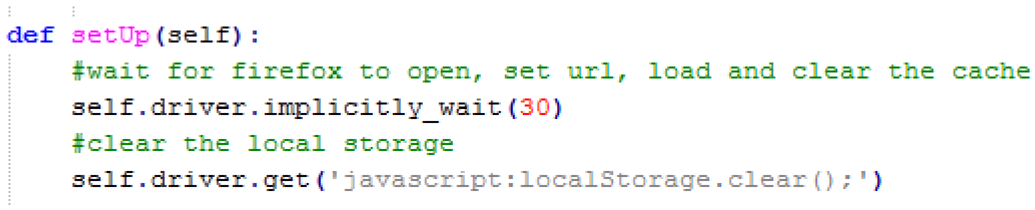
2.2 Test Script

Setup class used will be run once before the class testing begin. Declare the firefox driver to used for all testing process and set the web URL to test.



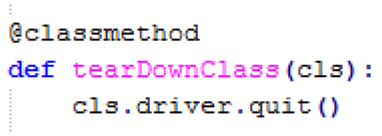
```
def setUpClass(cls):
    # create instance of Firefox driver
    cls.driver = webdriver.Firefox()
    #self.base_url = "http://todomvc.com/examples/react/"
    self.base_url = "http://todomvc.com/examples/angularjs"
```

Setup self will be run before executing each test. Wait 30 second for element to load, clear element cache.



```
def setUp(self):
    #wait for firefox to open, set url, load and clear the cache
    self.driver.implicitly_wait(30)
    #clear the local storage
    self.driver.get('javascript:localStorage.clear();')
```

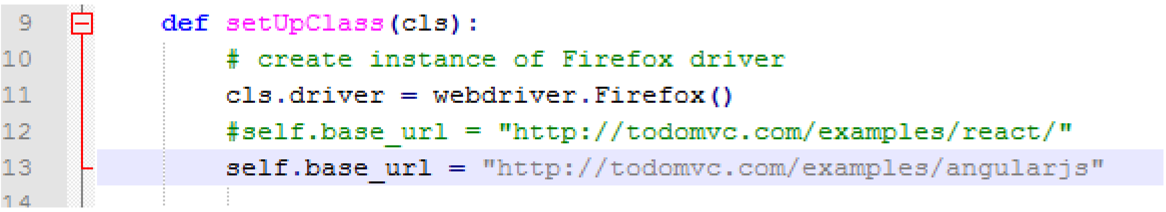
Tear down class will be called once before the testing class ended. Close the browser, before end the whole process.



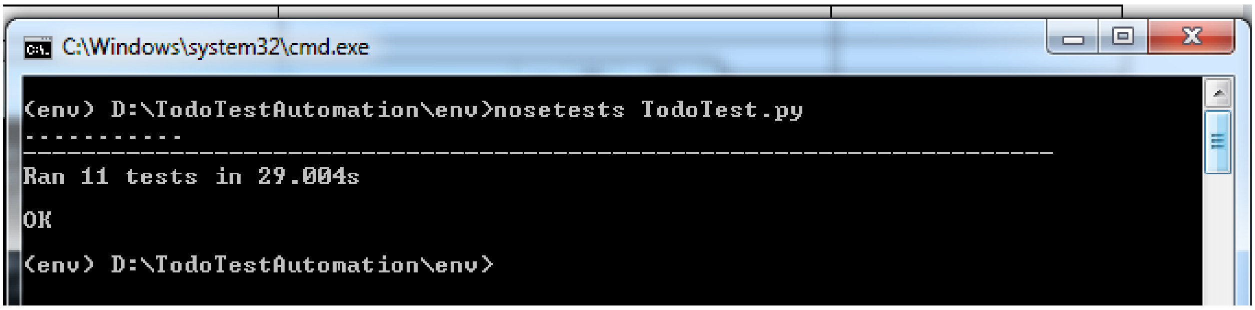
```
@classmethod
def tearDownClass(cls):
    cls.driver.quit()
```

2.3 Result

Run the testing for angularjs by insert command **nosetests TodoTest.py** . please make sure url set to angular.js (line 13)



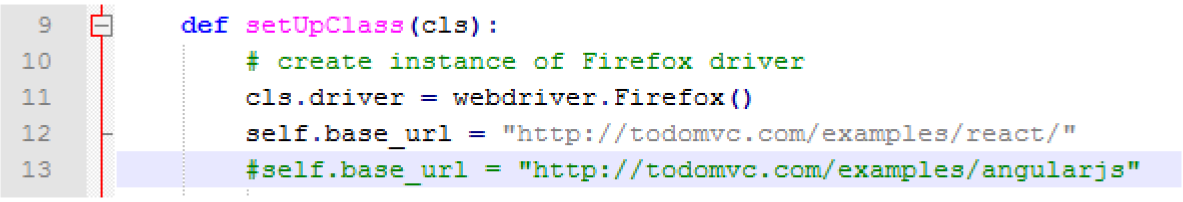
```
9      def setUpClass(cls):
10         # create instance of Firefox driver
11         cls.driver = webdriver.Firefox()
12         #self.base_url = "http://todomvc.com/examples/react/"
13         self.base_url = "http://todomvc.com/examples/angularjs"
14
```



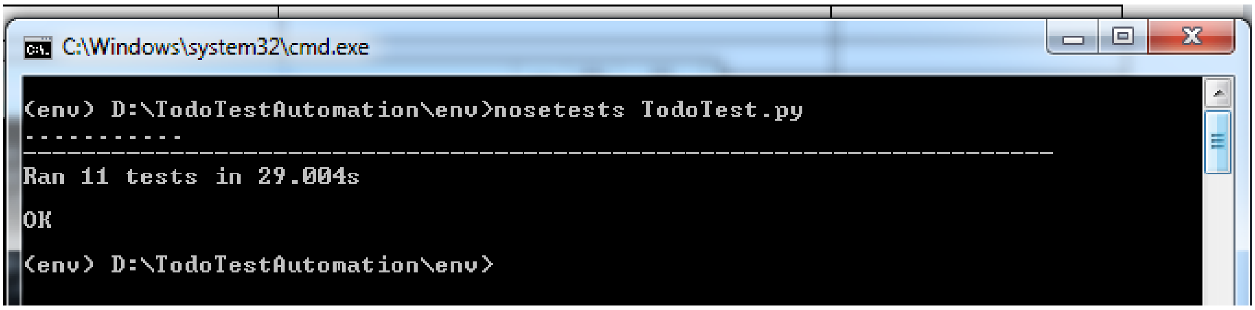
C:\Windows\system32\cmd.exe

```
(env) D:\TodoTestAutomation\env>nosetests TodoTest.py
...........
----------------------------------------------------------------------
Ran 11 tests in 29.004s

OK

(env) D:\TodoTestAutomation\env>
```

Run the testing for react change the URL by comment the line 13 and uncommand line12



```
    def setUpClass(cls):
        # create instance of Firefox driver
        cls.driver = webdriver.Firefox()
        self.base_url = "http://todomvc.com/examples/react/"
        #self.base_url = "http://todomvc.com/examples/angularjs"
```



```
(env) D:\TodoTestAutomation\env>nosetests TodoTest.py
...........
----------------------------------------------------------------------
Ran 11 tests in 29.004s

OK

(env) D:\TodoTestAutomation\env>
```

Source:
http://engineering.aweber.com/getting-started-with-ui-automated-tests-using-selenium-python/
http://selenium-python.readthedocs.org/locating-elements.html