Screen Shot in Selenium

Let’s see how to capture Screenshot in Python Selenium Webdriver

- Ability to take screenshots is most important and desirable feature for bug analysis.
- Sometimes, apart from bug analysis, one may want to capture screenshots to see the flow of test steps.
- Screenshots help automation testers a lot when test cases fails, one can identify what went wrong in test script or application.

Selenium can take screenshots during execution and save it in a file. Selenium webdriver has a methods to take screenshot and these methods support screenshot file as ‘.png’.

WebDriver offers total **three APIs** to take screenshot of a web page.

1. save_screenshot(‘filename’)
2. get_screenshot_as_file(‘filename’)
3. get_screenshot_as_png()

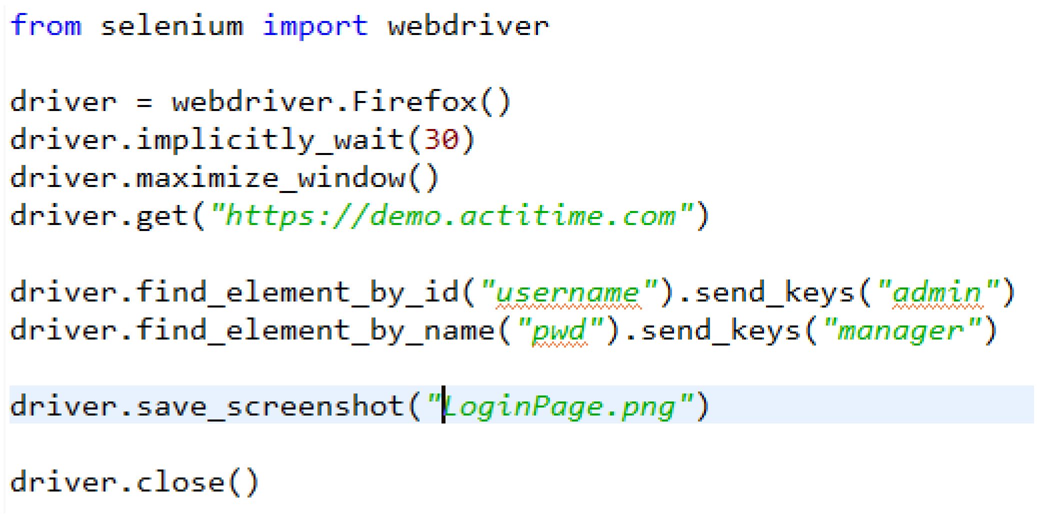
Note:

- First two APIs are used to take and store screenshots as ‘.png’ files.
- Third API, get_screenshot_as_png (), returns a binary data. This binary data will create an image in memory and can be useful if we want to manipulate before saving it.

Important: An important note to store screenshots is that save_screenshot(‘filename’) and get_screenshot_as_file(‘filename’) will work only when extension of file is ‘.png’ Otherwise content of the screenshot can’t be viewed and Python throws a warning message.

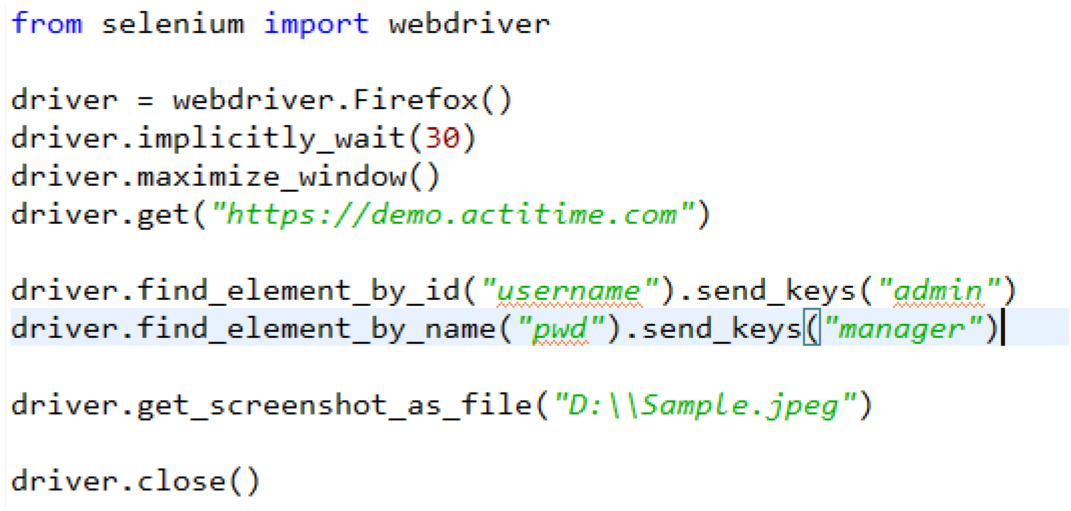
```
UserWarning: name used for saved screenshot does not match file type. It should end with a `.png` extension
“type. It should end with a `.png` extension”, UserWarning)
```

save_screenshot ()



```
from selenium import webdriver

driver = webdriver.Firefox()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://demo.actitime.com")

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")

driver.save_screenshot("LoginPage.png")

driver.close()
```

get_screenshot_as_file ()



```
from selenium import webdriver

driver = webdriver.Firefox()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://demo.actitime.com")

driver.find_element_by_id("username").send_keys("admin")
driver.find_element_by_name("pwd").send_keys("manager")

driver.get_screenshot_as_file("D:\\Sample.jpeg")

driver.close()
```

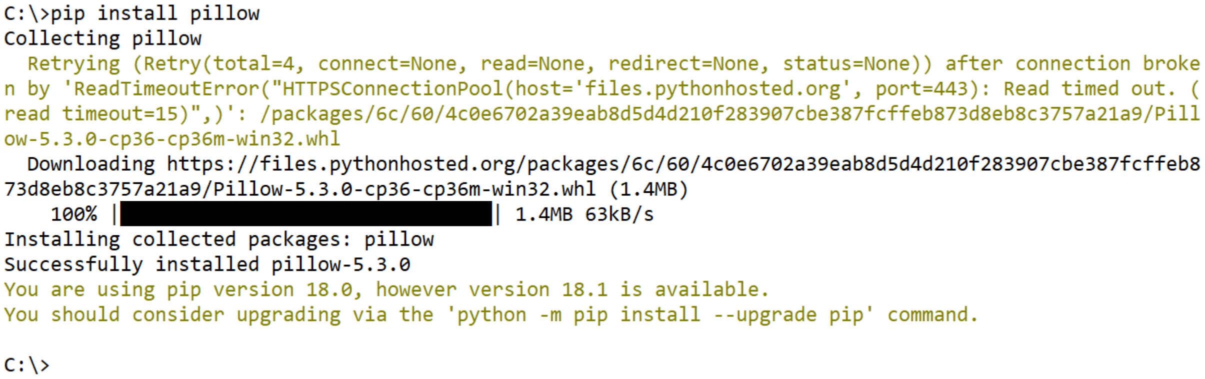
Capture screenshot of an Element using Python Selenium

- Sometimes we may just want to
    - Capture a part of the page,
    - Just on specific element based on ID, or
    - Any specific element locator.
      For example, we would like to capture the logo in Google page with id.
- Selenium WebDriver has feature only to capture the whole page; and does not have screenshot function that takes element id or name as input.
- Taking screenshot of an element is not straight forward in Selenium WebDriver.
- To do this, we have to capture screenshot of page first, get dimension and size of the element, and then using image libraries crop the image as required.

Step 1: Download the “**Pillow**” package.

Open command prompt enter the below command.

“Pip install pillow” – to install
“Pip install –U pillow” – to download latest version if pillow is already installed.



```
C:\>pip install pillow
Collecting pillow
  Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broke
n by 'ReadTimeoutError("HTTPSConnectionPool(host='files.pythonhosted.org', port=443): Read timed out. (
read timeout=15)",)': /packages/6c/60/4c0e6702a39eab8d5d4d210f283907cbe387fcffeb873d8eb8c3757a21a9/Pill
ow-5.3.0-cp36-cp36m-win32.whl
  Downloading https://files.pythonhosted.org/packages/6c/60/4c0e6702a39eab8d5d4d210f283907cbe387fcffeb8
73d8eb8c3757a21a9/Pillow-5.3.0-cp36-cp36m-win32.whl (1.4MB)
    100% |████████████████████████████████| 1.4MB 63kB/s
Installing collected packages: pillow
Successfully installed pillow-5.3.0
You are using pip version 18.0, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\>
```

Before proceeding screenshot, let’s look at two methods available for WebDriver element object.

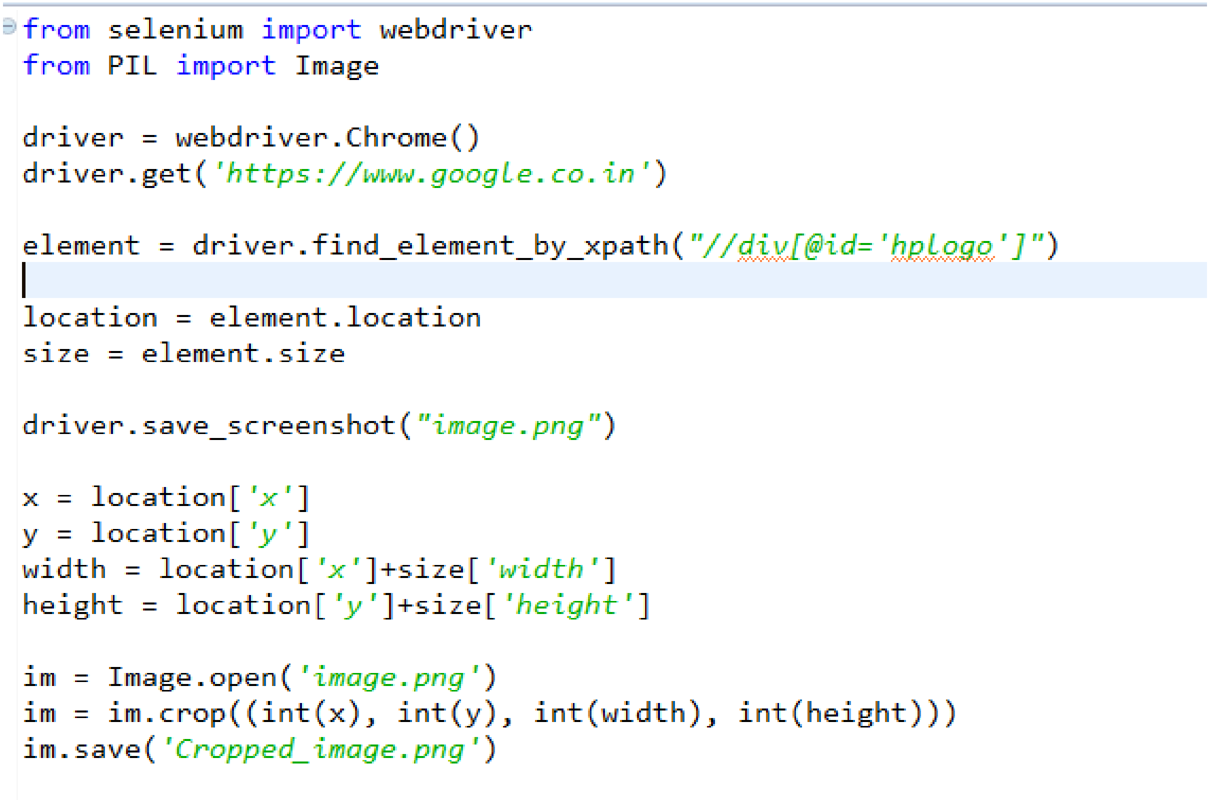
Location
A web element has its own position on page and generally it is measured in x and y pixels and known as (x, y) co-ordinates of element and location object contains these two values.

1. location[‘x’] returns ‘x’ co-ordinate of the element
2. location[‘y’] returns ‘y’ co-ordinate of the element

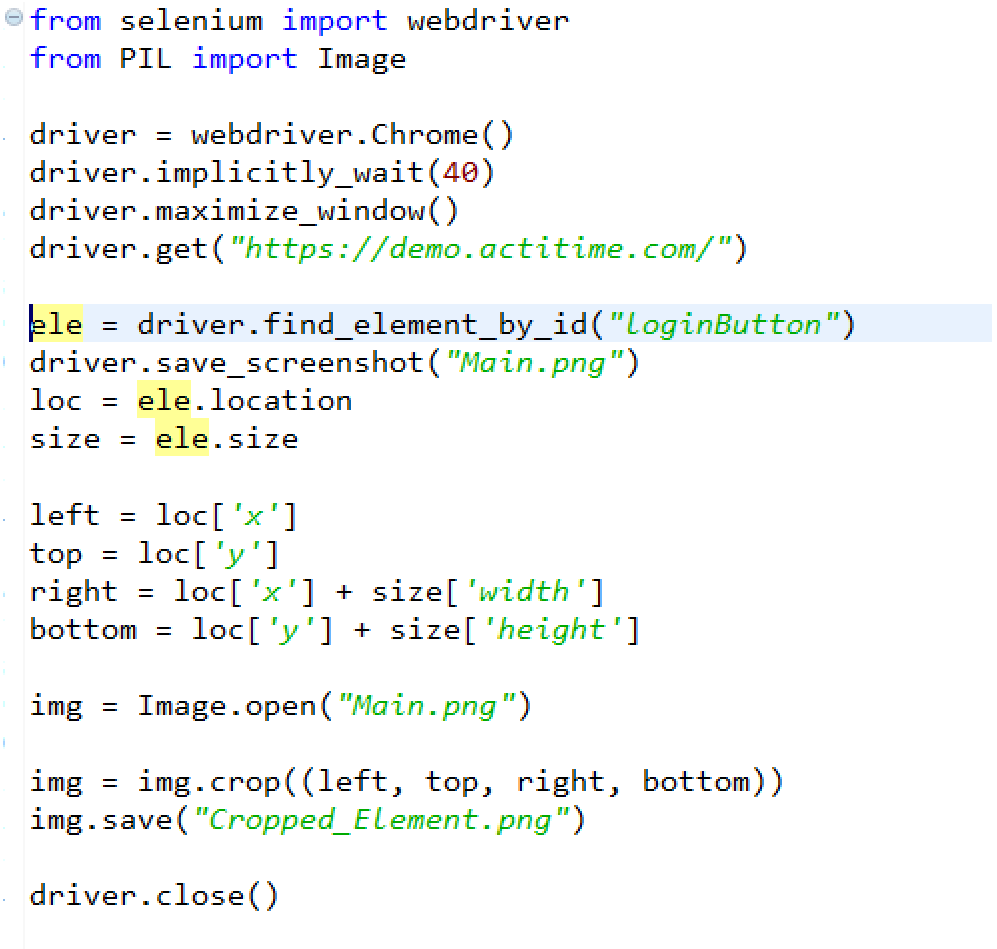
Size
Like location, each WebElement has width and height or size.

1. size[‘width’] returns ‘width’ of the element
2. size[‘height’] returns ‘height’ of the element

Using x, y co-ordinates and width, height values we can crop the image and store.



```
from selenium import webdriver
from PIL import Image

driver = webdriver.Chrome()
driver.get('https://www.google.co.in')

element = driver.find_element_by_xpath("//div[@id='hplogo']")

location = element.location
size = element.size

driver.save_screenshot("image.png")

x = location['x']
y = location['y']
width = location['x']+size['width']
height = location['y']+size['height']

im = Image.open('image.png')
im = im.crop((int(x), int(y), int(width), int(height)))
im.save('Cropped_image.png')
```



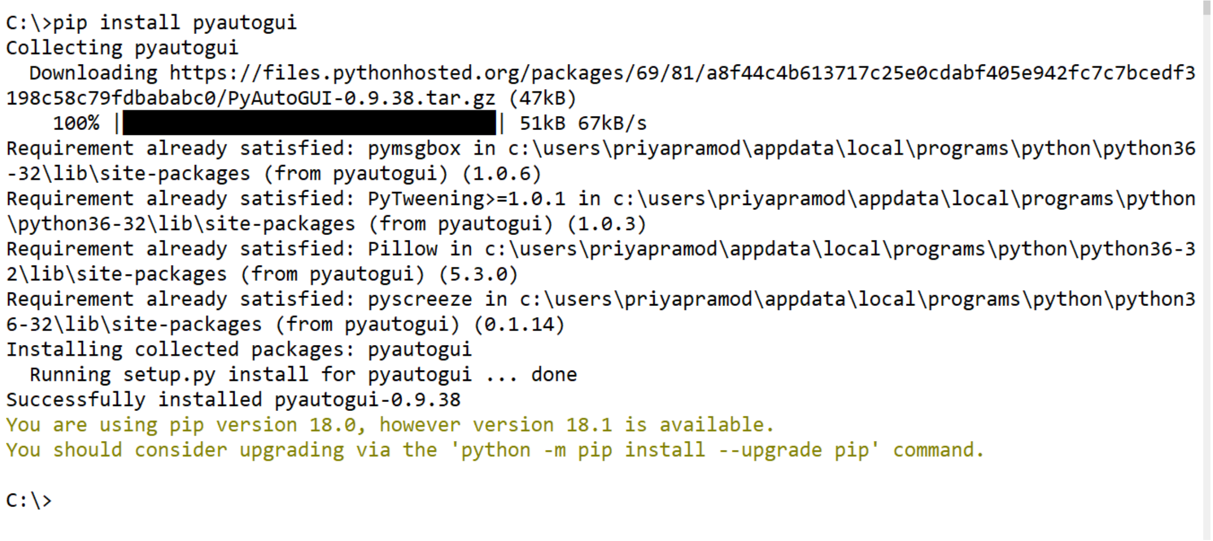
```
from selenium import webdriver
from PIL import Image

driver = webdriver.Chrome()
driver.implicitly_wait(40)
driver.maximize_window()
driver.get("https://demo.actitime.com/")

ele = driver.find_element_by_id("loginButton")
driver.save_screenshot("Main.png")
loc = ele.location
size = ele.size

left = loc['x']
top = loc['y']
right = loc['x'] + size['width']
bottom = loc['y'] + size['height']

img = Image.open("Main.png")

img = img.crop((left, top, right, bottom))
img.save("Cropped_Element.png")

driver.close()
```

Taking the screen shot of Non-HTML pop ups (Alert, file-upload, notification etc.)

- Selenium cannot take the screen shot of any non-html contents or pop ups.
- In order to take the screen shot of non-html pop ups, we use the package called "**PyAutoGUI**".

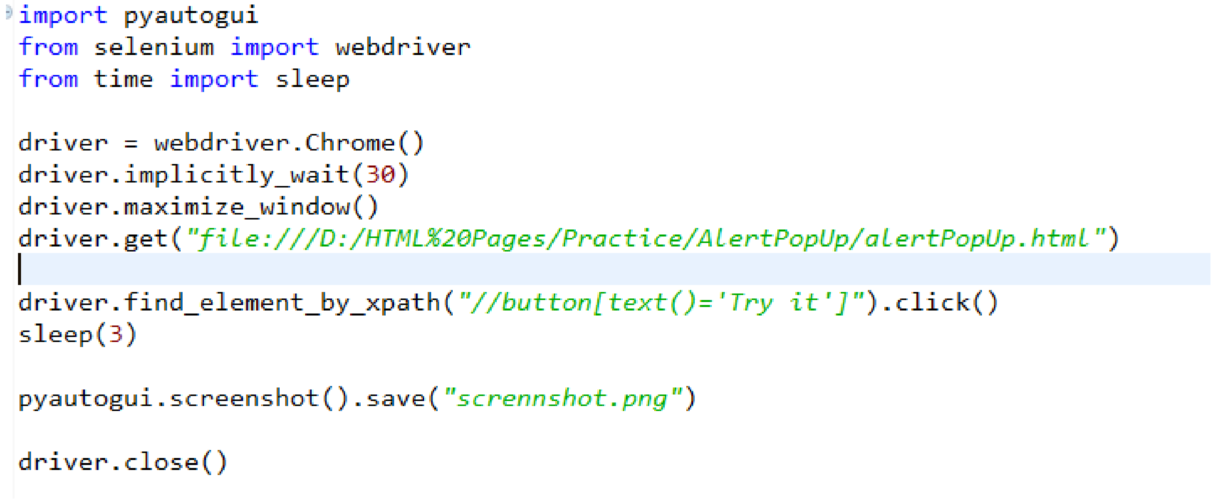
First, install the PyAutoGUI.
Open command prompt and type the following command.

"Pip install pyautogui"



```
C:\>pip install pyautogui
Collecting pyautogui
  Downloading https://files.pythonhosted.org/packages/69/81/a8f44c4b613717c25e0cdabf405e942fc7c7bcedf3
198c58c79fdbababc0/PyAutoGUI-0.9.38.tar.gz (47kB)
    100% |████████████████████████████████| 51kB 67kB/s
Requirement already satisfied: pymsgbox in c:\users\priyapramod\appdata\local\programs\python\python36
-32\lib\site-packages (from pyautogui) (1.0.6)
Requirement already satisfied: PyTweening>=1.0.1 in c:\users\priyapramod\appdata\local\programs\python
\python36-32\lib\site-packages (from pyautogui) (1.0.3)
Requirement already satisfied: Pillow in c:\users\priyapramod\appdata\local\programs\python\python36-3
2\lib\site-packages (from pyautogui) (5.3.0)
Requirement already satisfied: pyscreeze in c:\users\priyapramod\appdata\local\programs\python\python3
6-32\lib\site-packages (from pyautogui) (0.1.14)
Installing collected packages: pyautogui
  Running setup.py install for pyautogui ... done
Successfully installed pyautogui-0.9.38
You are using pip version 18.0, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\>
```

Program to take the screen shot of non-html pop up.



```
import pyautogui
from selenium import webdriver
from time import sleep

driver = webdriver.Chrome()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("file:///D:/HTML%20Pages/Practice/AlertPopUp/alertPopUp.html")

driver.find_element_by_xpath("//button[text()='Try it']").click()
sleep(3)

pyautogui.screenshot().save("scrennshot.png")

driver.close()
```