Automation testing

13.3.2023

Selenium is a library not a tool.

Watir is competitor of selenium.

Why we use automation:

1. Money/time
2. Reduce human error.

Mostly 8 pointers/html objects taken into consideration:

1. <p> = paragraph
2. H1…h6 = header
3. Img = logo/images
4. <a> = link
5. Button = link/button/input
6. Checkout/input
7. Radio,select.

Creating a x-path:

//AA = primary identifiers

//BB = secondary identifiers

//AA[text()=’full text’]// text only

//AA[container(@BB,’value of partial BB’)]--// non text

In logo:

AA= img

BB=src,alt,style

(//img[@alt='Webucator Logo'])[2]—This is an xpath created by us.

//input[@id='nav-search-mobile' and @placeholder='Search Courses'] // to concat two elements

//a [contains(.,'Browse Courses')]// 4 for partial text

//a [text()='Browse Courses ']// for full text we need to add spaces as well

//i[@class='ms-5 fa fa-solid fa-chevron-down open-nav-dropdown-icon']// element taken by class.

* Star health website:
* //header[@class='Header\_header-sticky\_\_heI5P Header\_header-show\_\_XGjmx']

//div[@data-index=4]//div/a//span[@class='a-price']

Amazon.in for redmi phones under 6k

Following sibling:

//div[@class='navFooterLinkCol navAccessibility']/div[text()='Let Us Help You']/following-sibling::ul/li/a

def Launch\_App(self):  
# driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()))  
global driver  
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  
driver.implicitly\_wait(10)  
#driver.get("[https://www.webucator.com/")](https://www.webucator.com/%22))  
#driver.get(Common\_method.readXmlAsPerNode("appUrl"))  
driver.maximize\_window()  
time.sleep(2)  
return self

def Check\_validateBox(self):  
  
try:  
#driver.get(Common\_method.readXmlAsPerNode("[https://the-internet.herokuapp.com/checkboxes"))](https://the-internet.herokuapp.com/checkboxes%22)))  
driver.get("[https://the-internet.herokuapp.com/checkboxes"](https://the-internet.herokuapp.com/checkboxes%22))  
driver.implicitly\_wait(10)  
time.sleep(2)  
# validation for 1 check box  
if not driver.find\_element(By.XPATH,"(//form[@id='checkboxes']/input)[1]").is\_selected():  
print("Checkbox 1 is not selected")  
driver.find\_element(By.XPATH, "(//form[@id='checkboxes']/input)[1]").click()  
time.sleep(2)  
else:  
print("Checkbox 1 is selected by default")  
  
# validation for 2 check box  
if driver.find\_element(By.XPATH,"(//form[@id='checkboxes']/input)[2]").is\_selected():  
print("Checkbox 2 is selected by default")  
driver.find\_element(By.XPATH, "(//form[@id='checkboxes']/input)[2]").click()  
if not driver.find\_element(By.XPATH, "(//form[@id='checkboxes']/input)[2]").is\_selected():  
print("Checkbox 2 is unchecked pass")  
time.sleep(2)  
  
time.sleep(2)  
else:  
print("Checkbox 1 is selected by default")  
except:  
print("Something went wrong")

def Sign\_in(self):  
driver.get("[https://www.webucator.com/account/login/"](https://www.webucator.com/account/login/%22))  
driver.implicitly\_wait(10)  
time.sleep(2)  
driver.find\_element(By.NAME,"login").click()  
driver.find\_element(By.NAME, "login").send\_keys("Subham@gmail.com")  
driver.find\_element(By.NAME, "password").click()  
driver.find\_element(By.NAME, "password").send\_keys("Shubham@123")  
time.sleep(3)  
#now clear screen...  
driver.find\_element(By.NAME, "login").click()  
driver.find\_element(By.NAME, "login").clear()  
driver.find\_element(By.NAME, "password").click()  
driver.find\_element(By.NAME, "password").clear()  
time.sleep(3)

driver.find\_element(By.NAME, "login").click()  
driver.find\_element(By.NAME, "login").clear()  
driver.find\_element(By.NAME, "password").click()  
driver.find\_element(By.NAME, "password").clear()  
time.sleep(3)

def Footer\_About(self):  
driver.get("[https://www.webucator.com/"](https://www.webucator.com/%22))  
driver.implicitly\_wait(10)  
coun=driver.find\_elements(By.XPATH,"//footer//h4[text()='About Us']/../ul/li/a")  
  
print("Total Links are",len(coun))  
for i in coun:  
print(i.text)  
time.sleep(2)

def Drop\_down(self):  
driver.get("[https://the-internet.herokuapp.com/dropdown"](https://the-internet.herokuapp.com/dropdown%22))  
driver.implicitly\_wait(10)  
select=Select(driver.find\_element(By.ID,'dropdown'))  
select.select\_by\_index(1);  
time.sleep(2)

def Handling\_Alert(self):  
driver.get("[https://the-internet.herokuapp.com/javascript\_alerts"](https://the-internet.herokuapp.com/javascript_alerts%22))  
driver.implicitly\_wait(10)  
time.sleep(2)  
driver.find\_element(By.XPATH,"//button[contains(.,'Click for JS Alert')]").click()  
time.sleep(2)  
alert = Alert(driver)  
if alert.text=="I am a JS Alert":  
alert.accept()  
time.sleep(1)  
driver.find\_element(By.XPATH,"//p[text()='You successfully clicked an alert']").is\_displayed()  
print("Click for Js Alert Pass.")  
else:  
print("Click for Js Alert Fail.")  
  
#Click for Js Confirmation.........  
driver.find\_element(By.XPATH, "//button[contains(.,'Click for JS Confirm')]").click()  
time.sleep(2)  
alert = Alert(driver)  
if alert.text == "I am a JS Confirm":  
alert.dismiss()  
time.sleep(2)  
driver.find\_element(By.XPATH, "//p[text()='You clicked: Cancel']").is\_displayed()  
print("Click for Js confim pass.")  
else:  
print("Click for Js confim fail.")  
  
# Click for Js Confirmation.........  
driver.find\_element(By.XPATH, "//button[contains(.,'Click for JS Prompt')]").click()  
time.sleep(2)  
alert = Alert(driver)  
if alert.text == "I am a JS prompt":  
alert.send\_keys("Dhoti")  
alert.accept()  
time.sleep(2)  
driver.find\_element(By.XPATH, "//p[text()='You entered: Dhoti']").is\_displayed()  
print("Click for Js Prompt With name pass.")  
else:  
print("Click for Js Prompt With namefail.")

def Attribute(self):  
driver.get("[https://www.docker.com/"](https://www.docker.com/%22))  
driver.implicitly\_wait(10)  
time.sleep(2)  
ele=driver.find\_element(By.XPATH,"//div[@class='os-item downloadable windows-stable']/a")  
print(ele.get\_attribute("href"))

16.3.23

pytest-> python-selenium testing framework