50. Selenium Hybrid Framework - 2

(Python, Selenium, PyTest, Page Object Model, HTML Reports)

Step 3: Automating Register account test case

3.1: Create page object classes for HomePage & AccountRegistration page under "pageObjects".

HomePage.py

```
from selenium.webdriver.common.by import By
class HomePage():
  lnk_myaccount_xpath = "//span[text()='My Account']"
  lnk_register_linktext = "Register"
  lnk_login_linktext = "Login"
  def __init__(self, driver):
     self.driver = driver
  def clickMyAccount(self):
     self.driver.find_element(By.XPATH, self.lnk_myaccount_xpath).click()
  def clickRegister(self):
     self.driver.find_element(By.LINK_TEXT,self.lnk_register_linktext).click()
  def clickLogin(self):
     self.driver.find_element(By.LINK_TEXT,self.lnk_login_linktext).click()
```

AccountRegistrationPage.py

```
from selenium.webdriver.common.by import By
class AccountRegistrationPage():
  txt_firstname_name = "firstname"
  txt_lastname_name = "lastname"
  txt_email_name = "email"
  txt_telphone_name = "telephone"
  txt_password_name = "password"
  txt_confpassword_name = "confirm"
  chk_policy_name = "agree"
  btn_cont_xpath="//input[@value='Continue']"
  text_msg_conf_xpath="//h1[normalize-space()='Your Account Has Been Created!']"
  def __init__(self, driver):
     self.driver = driver
  def setFirstName(self,fname):
    self.driver.find_element(By.NAME,self.txt_firstname_name).send_keys(fname)
  def setLastName(self,lname):
     self.driver.find_element(By.NAME,self.txt_lastname_name).send_keys(lname)
  def setEmail(self,email):
     self.driver.find_element(By.NAME,self.txt_email_name).send_keys(email)
  def setTelephone(self,tel):
    self.driver.find_element(By.NAME,self.txt_telphone_name).send_keys(tel)
  def setPassword(self,pwd):
     self.driver.find_element(By.NAME,self.txt_password_name).send_keys(pwd)
  def setConfirmPassword(self,cnfpwd):
     self.driver.find_element(By.NAME,self.txt_confpassword_name).send_keys(cnfpwd)
  def setPrivacyPolicy(self):
     self.driver.find_element(By.NAME,self.chk_policy_name).click()
  def clickContinue(self):
     self.driver.find_element(By.XPATH,self.btn_cont_xpath).click()
  def getconfirmationmsg(self):
     try:
        return  self.driver.find_element(By.XPATH,self.text_msg_conf_xpath).text
     except:
        None
```

3.2: Create conftest.py under "testCases" with driver

conftest.py

```
import pytest
from selenium import webdriver
@pytest.fixture()
def setup():
  options = webdriver.ChromeOptions()
  options.add_experimental_option("detach", True)
  driver = webdriver.Chrome(options=options)
  yield driver
  driver.quit()
```

3.3: Create AccountRegistration testcase under "testCases"

test_001_AccountRegistration.py

```
from pageObjects.HomePage import HomePage
from pageObjects.AccountRegistrationPage import AccountRegistrationPage
class Test_001_AccountReg:
  baseURL = "https://tutorialsninja.com/demo/"
  def test_account_reg(self,setup):
     self.driver = setup
     self.driver.get(self.baseURL)
     self.driver.maximize_window()
     self.hp=HomePage(self.driver)
     self.hp.clickMyAccount()
     self.hp.clickRegister()
     self.regpage=AccountRegistrationPage(self.driver)
     self.regpage.setFirstName("John")
     self.regpage.setLastName("Canedy")
     self.regpage.setEmail('test@gmail.com')
     self.regpage.setTelephone("65656565")
     self.regpage.setPassword("abcxyz")
     self.regpage.setConfirmPassword("abcxyz")
     self.regpage.setPrivacyPolicy()
     self.regpage.clickContinue()
     self.confmsg=self.regpage.getconfirmationmsg()
```

```
    if self.confmsg=="Your Account Has Been Created!":
        assert True
    else:
        assert False
```

Run tests on Terminal

- ➔ pytest -s -v .\testCases\test_001_AccountRegistration.py

3.4: Write a utility file to generate random string for email.

randomString.py

```
import random
import string
def random_string_generator(size=5, chars=string.ascii_lowercase + string.digits):
  return ''.join(random.choice(chars) for x in range(size))
```

Step 4: capture screenshot on failures

4.1: Update AccountRegistration Test case with capture Screenshot under "testCases"

test_001_AccountRegistration.py

```
import os
from pageObjects.HomePage import HomePage
from pageObjects.AccountRegistrationPage import AccountRegistrationPage
from utilities import randomString
class Test_001_AccountReg:
  baseURL = "https://tutorialsninja.com/demo/"
  def test_account_reg(self,setup):
    self.driver = setup
    self.driver.get(self.baseURL)
    self.driver.maximize_window()
    self.hp=HomePage(self.driver)
    self.hp.clickMyAccount()
    self.hp.clickRegister()
    self.regpage=AccountRegistrationPage(self.driver)
    self.regpage.setFirstName("John")
    self.regpage.setLastName("Canedy")
    self.email=randomString.random_string_generator()+'@gmail.com'
    self.regpage.setEmail(self.email)
    self.regpage.setTelephone("65656565")
    self.regpage.setPassword("abcxyz")
    self.regpage.setConfirmPassword("abcxyzz")
    self.regpage.setPrivacyPolicy()
    self.regpage.clickContinue()
    self.confmsg=self.regpage.getconfirmationmsg()
    if self.confmsg=="Your Account Has Been Created!":
            assert True
    else:
        self.driver.save_screenshot(os.path.dirname(os.getcwd()) + "\\screenshots\\" +
"test_account_reg.png")    #from IDE
```

or   #from Terminal

```
self.driver.save_screenshot(os.path.abspath(os.curdir) + "\\screenshots\\" + "test_account_reg.png")
        assert False
```

Normally when we work selenium with java combination we will have a **properties files**..Properties files are more compatible with java.Similarly **.ini** file is more compatible with python.This file contains some configurations and what are all **common data (base url,email address,password)** required for test case we will put the data in the ini file and we will read the data from the ini file

i.e, if we have multiple test cases instead of entering email id,password everywhere in every test case we will create a separate file .init file contains url of application ,email address password and in every test case we will get the data from that file. Again we cannot directly represent this ini file but in between we should create some utility file which will get the data from the .ini file and provide the same data to all the test cases.

Advantage

In future if the data is changed and we no need to modify every test case we have to just modify the single file .ini file and that will automatically be reflected in every test case because we are not hard coding data in every test case

Step 5: Read common values from ini file.

5.1: Add "config.ini" file in the "configurations" folder.

config.ini file

```
[commonInfo]
baseURL = https://tutorialsninja.com/demo/
email=abcxyz@gmail.com
password=abcxyz
```

5.2: Create "readProperties.py" utility file under utilities package to read common data.

readProperties.py

```
import configparser
import os
config = configparser.RawConfigParser()
config.read(os.path.dirname(os.getcwd())+'\\OpenCart\\configurations\\config.ini')
class ReadConfig():
  @staticmethod
  def getApplicationURL():
      url=(config.get('commonInfo', 'baseURL'))
      return url
  @staticmethod
  def getUseremail():
      username=(config.get('commonInfo', 'email'))
      return username
  @staticmethod
  def getPassword():
      password=(config.get('commonInfo', 'password'))
```

```
    return password
#Testing above methods - optional Code
print(ReadConfig.getApplicationURL())
print(ReadConfig.getUseremail())
```

5.3: Replace hard coded values in AccountRegistration testcase.

test_001_AccountRegistration.py

```
import os.path
from pageObjects.HomePage import HomePage
from pageObjects.AccountRegistrationPage import AccountRegistrationPage
from utilities.readProperties import ReadConfig
class Test_001_AccountReg:
  baseURL = ReadConfig.getApplicationURL()
  def test_account_reg(self,setup):
    self.driver = setup
    self.driver.get(self.baseURL)
    self.driver.maximize_window()
    self.driver.implicitly_wait(10)
    self.hp = HomePage(self.driver)
    self.hp.clickMyAccount()
    self.hp.clickRegister()
    self.regpage = AccountRegistrationPage(self.driver)
    self.regpage.setFirstName("John")
    self.regpage.setLastName("Canedy")
    self.regpage.setEmail(ReadConfig.getUseremail())
    self.regpage.setTelephone("65656565")
    self.regpage.setPassword(ReadConfig.getPassword())
    self.regpage.setConfirmPassword(ReadConfig.getPassword())
    self.regpage.setPrivacyPolicy()
    self.regpage.clickContinue()
    self.confmsg = self.regpage.getconfirmationmsg()
    if self.confmsg == "Your Account Has Been Created!":
      assert True
    else:
      self.driver.save_screenshot(os.path.abspath(os.curdir) + "\\screenshots\\" +
"test_account_reg.png")
      assert False
```