**SELENIUM\_04**

**01 TESTCASE**

1. **login\_testcase**

\_\_author\_\_ = 'Administrator'

import unittest

from selenium import webdriver

from time import sleep

from selenium\_day3.user\_login.def\_user\_login import login\_by\_different\_username\_password

from selenium\_day3.shijianchuo.def\_shijianchuo import jietu

# 定义一个类，这个类必须继承自unittest.TestCase这个类

class UserLoginTestCase(unittest.TestCase):

# 定义setUp这个前置方法，执行用例之前的公共操作

def setUp(self):

# 驱动浏览器，设置最大化

self.browser = webdriver.Firefox()

self.browser.maximize\_window()

# 定义tearDown这个后置方法，执行用例之后的公共操作

def tearDown(self):

# 关闭浏览器

self.browser.quit()

# 定义方法，方法名字以test\_开头，实现正常场景登录，登录完成后使用assertEqual进行断言

def test\_01\_login\_by\_correct\_username\_password(self):

try:

login\_by\_different\_username\_password(self.browser,"sup","s1234567")

sleep(2)

# 使用登录后的用户名进行断言

username\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/table/tbody/tr/td[2]/table/tbody/tr[3]/td[1]/label')

username = username\_ele.text

self.assertEqual(username,"苏鹏")

except Exception as e:

jietu(self.browser,"登录异常")

raise e

# 定义方法，使用正确的用户名和错误的密码登录，登录完成后提示错误信息使用assertEqual进行断言

def test\_02\_login\_by\_correct\_username\_wrong\_password(self):

try:

login\_by\_different\_username\_password(self.browser,"sup","s1234567888")

sleep(2)

# 使用错误提示信息的内容进行断言

tips\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')

tips = tips\_ele.text

self.assertEqual("登录名或密码错误，请重试！",tips)

except Exception as e:

jietu(self.browser,"登录异常")

raise e

# 定义方法，使用错误的用户名和正确的密码登录，登录完成后提示错误信息使用assertEqual进行断言

def test\_03\_login\_by\_wrong\_username\_correct\_password(self):

try:

login\_by\_different\_username\_password(self.browser,"supppp","s1234567")

sleep(2)

# 使用错误提示信息的内容进行断言

tips\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')

tips = tips\_ele.text

self.assertEqual("登录名或密码错误，请重试！",tips)

except Exception as e:

jietu(self.browser,"登录异常")

raise e

# 定义方法，使用正确的用户名空的密码进行登录，完成后使用是否存在提示信息assertIsNotNone进行断言

def test\_04\_login\_by\_correct\_username\_null\_password(self):

try:

login\_by\_different\_username\_password(self.browser,"sup","")

sleep(2)

tips\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')

self.assertIsNotNone(tips\_ele)

except Exception as e:

jietu(self.browser,"登录异常")

raise e

# 定义方法，使用空的用户名正确的密码进行登录，完成后使用是否存在提示信息assertIsNotNone进行断言

def test\_05\_login\_by\_null\_username\_correct\_password(self):

try:

login\_by\_different\_username\_password(self.browser,"","s1234567")

sleep(2)

tips\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')

self.assertIsNotNone(tips\_ele)

except Exception as e:

jietu(self.browser,"登录异常")

raise e

# 定义方法，假定登录按钮不可点击，输入用户名密码后获取登录按钮的点击状态，使用assertTrue进行断言

def test\_06\_login\_button\_clickable(self):

try:

self.browser.get("http://172.21.5.51:8080/oa")

sleep(2)

# 输入用户名

self.browser.find\_element\_by\_xpath('//input[@class="loginInput"]').clear()

self.browser.find\_element\_by\_xpath('//input[@class="loginInput"]').send\_keys("sup")

# 输入密码

self.browser.find\_element\_by\_xpath('//input[@name="password"]').send\_keys("s1234567")

# 获取登录按钮的状态

button\_status = self.browser.find\_element\_by\_xpath('//input[@id="button\_submit"]').is\_enabled()

self.assertTrue(button\_status)

except Exception as e:

jietu(self.browser,"登录异常")

raise e

if \_\_name\_\_=="\_\_main\_\_":

unittest.TextTestRunner()

1. **change\_password**

\_\_author\_\_ = 'Administrator'

import unittest

from selenium import webdriver

from time import sleep

from selenium\_day3.user\_login.def\_user\_login import login\_by\_different\_username\_password

from selenium\_day3.shijianchuo.def\_shijianchuo import jietu

from selenium\_day3.change\_password.def\_change\_password import change\_password

class ChangePasswordTestCase(unittest.TestCase):

def setUp(self):

self.browser = webdriver.Firefox()

self.browser.maximize\_window()

login\_by\_different\_username\_password(self.browser,"sup","s1234567")

sleep(2)

def tearDown(self):

self.browser.quit()

def test\_01\_wrong\_old\_password(self):

try:

change\_password(self.browser,"s123123123","s1111111","s1111111")

sleep(2)

# 断言

tips = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]').text

self.assertEqual("旧密码输入错误！",tips)

except Exception as e:

jietu(self.browser,"修改密码报错")

raise e

def test\_02\_different\_new\_password(self):

try:

change\_password(self.browser,"s1234567","s1111111","s2222222")

sleep(2)

# 断言

tips\_ele = self.browser.find\_element\_by\_xpath('/html/body/div[1]/div/table/tbody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')

self.assertIsNotNone(tips\_ele)

except Exception as e:

jietu(self.browser,"修改密码报错")

raise e

unittest.TextTestRunner()

**02\_testsuit**

1. **addtest**

\_\_author\_\_ = 'Administrator'

import unittest

from selenium\_day4.testcase.login\_testcase import UserLoginTestCase

def TestCase\_to\_suit\_by\_addtest():

# 实例化unittest下的testsuit类的对象

testcasesuit = unittest.TestSuite()

# 使用testcasesuit对象调用addtest的方法，将类中的用例一个一个添加到套件中

testcasesuit.addTest(UserLoginTestCase("test\_01\_login\_by\_correct\_username\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_02\_login\_by\_correct\_username\_wrong\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_03\_login\_by\_wrong\_username\_correct\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_04\_login\_by\_correct\_username\_null\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_05\_login\_by\_null\_username\_correct\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_06\_login\_button\_clickable"))

return testcasesuit

if \_\_name\_\_=="\_\_main\_\_":

# 实例化unittest下的TextTestRunner类的对象

test\_runner = unittest.TextTestRunner()

# 获取测试套件

test\_suit = TestCase\_to\_suit\_by\_addtest()

# 通过test\_runner调用run的方法来执行套件

test\_runner.run(test\_suit)

1. **makesuit**

\_\_author\_\_ = 'Administrator'

import unittest

from selenium\_day4.testcase.login\_testcase import UserLoginTestCase

def TestCase\_to\_suit\_by\_makesuit():

# 调用unittest下的makesuit的方法将类中的用例打包

class\_testcase = unittest.makeSuite(UserLoginTestCase)

# 调用unittest下的testsuit的方法获取套件并将打包好的用例添加到套件中

test\_suit = unittest.TestSuite(class\_testcase)

return test\_suit

if \_\_name\_\_ == "\_\_main\_\_":

test\_runner = unittest.TextTestRunner()

test\_suit = TestCase\_to\_suit\_by\_makesuit()

test\_runner.run(test\_suit)

**3) discover**

\_\_author\_\_ = 'Administrator'

import unittest

def TestCase\_to\_suit\_by\_discover():

# 确定用例存放的路径

testcase\_path = "D:\\zhaojia\\selenium\\selenium\_105\\selenium\_study\\selenium\_day4\\testcase\\"

# 调用unittest下的defaultTestLoader方法下的discover方法，来获取测试用例并添加到测试套件中

test\_suit = unittest.defaultTestLoader.discover(testcase\_path,pattern="\*.py")

return test\_suit

if \_\_name\_\_=="\_\_main\_\_":

test\_runner = unittest.TextTestRunner()

test\_suit = TestCase\_to\_suit\_by\_discover()

test\_runner.run(test\_suit)

1. **report**

\_\_author\_\_ = 'Administrator'

import sys

sys.path.append("D:\\zhaojia\\selenium\\selenium\_105\\selenium\_study")

import HTMLTestRunner

from selenium\_day4.testsuit.discover import TestCase\_to\_suit\_by\_discover

from selenium\_day3.shijianchuo.def\_shijianchuo import shijianchuo

# 确定测试报告存放的路径、名字、格式

HTMLfile = "D:\\zhaojia\\selenium\\selenium\_105\\selenium\_study\\report\\HTMLreport" + shijianchuo() + ".html"

# 打开文件并准备写入

htmlreport\_file = open(HTMLfile,"wb") # wb表示二进制的方式打开并写入

# 获取测试套件，获取执行的用例

test\_suit = TestCase\_to\_suit\_by\_discover()

# 实例化HTMLTestRunner下的HTMLTestRunner对象，需要配置参数，写清楚报告的标题、描述内容

htmlrunner = HTMLTestRunner.HTMLTestRunner(stream=htmlreport\_file,title="OA测试报告",description="共计8条用例，执行结果如下：")

# 使用runner的对象来调用run的方法，执行测试套件

htmlrunner.run(test\_suit)

# 关闭文件

htmlreport\_file.close()