import time

from unittest import TestCase

from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.by import By

from webdriver\_manager.chrome import ChromeDriverManager

class TestAlerts(TestCase):

driver = None

LINK = "https://the-internet.herokuapp.com/javascript\_alerts"

BUTTON\_JS\_ALERT\_SIMPLE = (By.CSS\_SELECTOR, "[onclick='jsAlert()']") # XPATH: //button[@onclick='jsAlert()']

BUTTON\_JS\_ALERT\_CONFIRM = (By.CSS\_SELECTOR, "[onclick='jsConfirm()']") # XPATH: //button[@onclick='jsConfirm()']

BUTTON\_JS\_ALERT\_PROMPT = (By.XPATH, "//button[contains(text(),'JS Prompt')]")

P\_RESULT = (By.ID, "result")

# suprascriem metoda setUp care va rula inainte de fiecare test

def setUp(self):

self.service = Service(ChromeDriverManager().install())

self.driver = webdriver.Chrome(service=self.service)

self.driver.get(self.LINK)

self.driver.maximize\_window()

time.sleep(1)

self.driver.implicitly\_wait(1)

# suprascriem metoda tearDown care va rula dupa fiecare test

def tearDown(self):

self.driver.quit()

# Metoda ajutatoare (nu se ruleaza ca test) -> ne ajuta sa nu mai scriem self.driver.find\_element(\*locator).click()

# la fiecare click, ci mai sumplu self.click(self.NUME\_LOCATOR)

def click(self, locator):

self.driver.find\_element(\*locator).click()

def test\_accept\_simple\_alert(self):

self.click(self.BUTTON\_JS\_ALERT\_SIMPLE)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

# apasam butonul care accepta alerta

alert.accept()

expected\_text = "You successfully clicked an alert"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

# assert expected\_text == actual\_text, f"Error, texts are not matching. \n" \

# f"Expected: {expected\_text} \n" \

# f"Actual: {actual\_text}"

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_accept\_confirm\_alert(self):

self.click(self.BUTTON\_JS\_ALERT\_CONFIRM)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

# apasam butonul care accepta alerta

alert.accept()

expected\_text = "You clicked: Ok"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_cancel\_confirm\_alert(self):

self.click(self.BUTTON\_JS\_ALERT\_CONFIRM)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

# apasam butonul care respinge alerta

alert.dismiss()

expected\_text = "You clicked: Cancel"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_accept\_prompt\_alert\_without\_text(self):

self.click(self.BUTTON\_JS\_ALERT\_PROMPT)

alert = self.driver.switch\_to.alert

alert.accept()

expected\_text = "You entered:"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_accept\_prompt\_alert\_with\_text(self):

self.click(self.BUTTON\_JS\_ALERT\_PROMPT)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

inserted\_text = "test"

alert.send\_keys(inserted\_text)

# apasam butonul care accepta alerta

alert.accept()

expected\_text = "You entered: " + inserted\_text

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_cancel\_prompt\_alert\_without\_text(self):

self.click(self.BUTTON\_JS\_ALERT\_PROMPT)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

alert.dismiss()

expected\_text = "You entered: null"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")

def test\_cancel\_prompt\_alert\_with\_text(self):

self.click(self.BUTTON\_JS\_ALERT\_PROMPT)

# declaram variabila pentru alerta, ca sa putem interactiona cu ea

alert = self.driver.switch\_to.alert

inserted\_text = "test"

alert.send\_keys(inserted\_text)

alert.dismiss()

expected\_text = "You entered: null"

actual\_text = self.driver.find\_element(\*self.P\_RESULT).text

self.assertEquals(expected\_text, actual\_text, "Error, texts are not matching.")