Selenium Automation For Dropdowns & Radio Buttons Select

This document provides an explanation of a Selenium script that interacts with a Google Form, filling out text fields, selecting radio buttons, choosing dropdown options, and finally submitting the form.

Table of Contents

1.Setup and Initialization

2.Navigating to Google Forms

3.Form Interactions

- Text Fields
- Radio Buttons
- Dropdowns

4.Form Submission

5.Cleanup and Closure

1.Setup and Initialization

The script starts by importing the required modules and setting up the Selenium WebDriver with Chrome. It also defines an instance of WebDriverWait, allowing the script to wait until certain conditions are met before proceeding.

code

```
import time

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait
```

```
from selenium.webdriver.support import expected_conditions as EC

from webdriver_manager.chrome import ChromeDriverManager


driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  # Initialize Chrome WebDriver

wait = WebDriverWait(driver, 10)  # Define a 10-second explicit wait
```

WebDriver Initialization: This sets up the Chrome WebDriver, allowing Selenium to control the browser.

Explicit Wait: The WebDriverWait instance is used to ensure elements are ready before interacting with them.

2.Navigating to Google Forms

The script navigates to a specific Google Form using its URL. It includes a delay to ensure the page has time to load before further interactions.

code

```
driver.get("https://docs.google.com/forms/d/e/1FAIpQLSeI8_vYyaJgM7SJM4Y9AWfLq-tglWZh6yt7bEXEOJr_L-hV1A/viewform?formkey=dGx0b1ZrTnoyZDgtYXItMWVBdVlQQWc6MQ")

time.sleep(5)
```

3.Form Interactions

This section outlines the various interactions with form elements, including text fields, radio buttons, dropdowns, and the final form submission.

Text Fields :

The script interacts with text fields using Selenium's find_element and send_keys to simulate user input. It uses explicit waits to ensure the elements are located and ready for interaction.

code

```
# Locate the first name field and send text
```

```
first_name = driver.find_element(By.XPATH,
'//*[@id="mG61Hd"]/div[2]/div/div[2]/div[1]/div/div/div[2]/div/div[1]/div/div[1]/input')

first_name.send_keys("Hello working fine")



# Locate the last name field and send text

last_name = driver.find_element(By.XPATH,
'//*[@id="mG61Hd"]/div[2]/div/div[2]/div[2]/div/div/div[2]/div/div[1]/div[2]/textarea')

last_name.send_keys("selenium automation testing")
```

Finding Elements: The find_element method is used to locate elements by XPath.

Sending Text: The send_keys method simulates typing into a field.

Delays Between Actions: A time.sleep(2) delay is added to give the browser time to process interactions.

Radio Buttons :

To interact with radio buttons, the script uses explicit waits to ensure the elements are clickable and then clicks them to select the desired option. Scrolling into view ensures the elements are visible before interaction.

code

```
# Locate and click a radio button

radio_button = wait.until(

   EC.element_to_be_clickable((By.XPATH, '...'))  # XPath for the radio button

)

driver.execute_script("arguments[0].scrollIntoView(true);", radio_button)  # Ensure visibility

radio_button.click()  # Click the radio button
```

Explicit Waits: The element_to_be_clickable condition ensures the radio button is interactable.

Scroll Into View: Using execute_script("arguments[0].scrollIntoView(true);", element) brings the element into view.

Clicking Radio Button: Once the radio button is visible and clickable, it's clicked to select it.

Dropdowns :

Dropdowns in Google Forms may require specific steps to interact with them. This includes clicking the dropdown trigger to open the list of options, then selecting a specific option.

code

```
# Open the dropdown

dropdown_trigger = wait.until(

   EC.element_to_be_clickable((By.XPATH, '...'))  # XPath for the dropdown trigger

)

dropdown_trigger.click()  # Open the dropdown



# Select an option from the dropdown

dropdown_option = wait.until(

   EC.element_to_be_clickable((By.XPATH, '...'))  # XPath for the dropdown option

)

dropdown_option.click()  # Click to select the option
```

Open the Dropdown: Click the dropdown trigger to open the list of options.

Select an Option: The element_to_be_clickable condition is used to ensure the dropdown option is interactable before clicking it.

Scroll Into View: If the dropdown is lower on the page, ensure it's visible by scrolling into view.

4.Form Submission

After filling out all required form elements, the script clicks the submit button to send the form.

code

```
submit = wait.until(

   EC.element_to_be_clickable((By.XPATH, '...'))  # XPath for the submit button

)

driver.execute_script("arguments[0].scrollIntoView(true);", submit)  # Ensure visibility
```

submit.click() # Click to submit the form

Explicit Waits for Submission: Ensure the submit button is clickable before clicking.

Scroll Into View: Make sure the submit button is visible before interaction.

5.Cleanup and Closure

After submitting the form, the script waits to ensure the form submission completes and then closes the browser to clean up resources.

code

time.sleep(10) # Allow time for the form to submit

driver.close() # Close the browser

Wait for Submission: The time.sleep(10) gives the form time to process the submission.

Close the Browser: Closing the WebDriver ensures that resources are released.