WhatsApp Automation Using Selenium - Step-by-Step Code Explanation

This script uses Selenium to automate sending a WhatsApp message via WhatsApp Web. It simulates browser actions like opening the site, searching a contact, and sending a message.

from selenium import webdriver

Import the Selenium WebDriver module to control the browser.

from selenium.webdriver.chrome.service import Service

Import the Service class to manage ChromeDriver service.

from webdriver\_manager.chrome import ChromeDriverManager

Import ChromeDriverManager to auto-manage the ChromeDriver binary.

from selenium.webdriver.common.by import By

Import By for locating elements using different strategies (e.g., XPATH).

from selenium.webdriver.common.keys import Keys

Import Keys to simulate keyboard inputs like ENTER.

from selenium.webdriver.support.ui import WebDriverWait

Import WebDriverWait to wait until elements are loaded or clickable.

from selenium.webdriver.support import expected\_conditions as EC

Import expected\_conditions to define element load conditions.

import time

Import time for delays (e.g., sleep between actions).

contact = "Shreya Gupta Roommate Pg"

Set the name of the contact you want to send the message to.

text = "Hey, this message was sent using Selenium"

Define the message text to be sent.

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

Initialize the Chrome WebDriver using ChromeDriverManager.

driver.get("https://web.whatsapp.com")

Open WhatsApp Web in a Chrome browser.

print("Scan the QR code and press Enter when done...")

Prompt user to scan the QR code for login.

input("Press Enter after scanning QR code...")

Pause script until user presses Enter after scanning the QR code.

print("Logged In")

Print confirmation once logged in.

inp\_xpath\_search = "//div[@title='Search input textbox']"

Define XPath to locate the WhatsApp search box.

WebDriverWait(driver, 60).until(EC.presence\_of\_element\_located((By.XPATH, inp\_xpath\_search)))

Wait until the search box is loaded on the page.

search\_box = driver.find\_element(By.XPATH, inp\_xpath\_search)

Locate the search box using the defined XPath.

search\_box.click()

Click the search box to activate it.

search\_box.send\_keys(contact)

Type the contact name into the search field.

time.sleep(2)

Pause briefly to allow search results to populate.

selected\_contact = WebDriverWait(driver, 30).until(  
 EC.presence\_of\_element\_located((By.XPATH, f"//span[@title='{contact}']"))  
)

Wait until the specified contact appears in the search results.

selected\_contact.click()

Click on the desired contact to open the chat.

msg\_box\_xpath = "//div[@title='Type a message']"

Define XPath to locate the message input box.

input\_box = WebDriverWait(driver, 30).until(EC.presence\_of\_element\_located((By.XPATH, msg\_box\_xpath)))

Wait until the message box is loaded.

input\_box.send\_keys(text + Keys.ENTER)

Type the message followed by ENTER to send it.

print("✅ Message sent!")

Print confirmation that the message was sent.

time.sleep(2)

Pause before closing the browser.

driver.quit()

Close the browser to end the session.