Technical Documentation: Automated User Signup - Authorized Partner

Demo Link: https://www.loom.com/share/60eaf0b38e834e108fce627d26da97d5

1. Project Overview

This automation package is aimed to conduct an end-to-end (E2E) functional test of Authorized Partner signup portal. The script automates the whole registration process, as well as a complicated integration with the Gmail API (via IMAP) to bypass Multi-Factor Authentication (MFA) by requesting a 6-digit OTP over the series of a live request.

2. Technical Stack

- Language: Java
- Automation Tool: Selenium WebDriver (Chrome)
- Email Protocol: JavaMail API (IMAPS)
- Wait Mechanism: Explicit Waits (WebDriverWait)
- Pattern Matching: Regex (java.util.regex)

3. Key Functionalities

A. Real-Time OTP Extraction

The application has a powerful email management application. It links to a Gmail inbox of the user and uses an App Password to search unread messages and manipulates Regular Expressions to decode the 6-digit verification code.

B. Stale Element Handling

A safeClick helper technique is adopted to make sure that the script is stable. This approach tries to tap an element until it has been clicked 3 times in case of a

StaleElementReferenceException (which is frequent in modern JavaScript applications) happens.

C. Multi-Step Form Automation

In the script, the registration takes place in four separate stages:

- Consent and Personal Data: Dealing with checkboxes and foundational user information.
- OTP Verification: Closing the gap between the web interface and the email server.
- Agency Details: Dropdown (comboboxes) and geographic selections should be automated.
- Document Upload: Streamlining the registration and certification PDF file selection.

4. Code Structure & Logic

- **Modular Logic and Resiliency**

The script starts with the utility functions meant to take care of the frequent automation failures. The safeClick technique is a self-healing tool, which relies on a retry loop to click the elements regardless of whether the webpage is automatically repopened. Likewise, the email processing is broken down into two sections: one to connect with the Gmail server securely and the other to search through the complicated email data with the help of the Regular Expressions to locate the exact 6-digit code.

- **Execution Workflow**

The primary mode is the engine, as it causes the browser to go through in a sequential, step-by-step process. It initially takes care of the original UI navigation and data entry and then halts until it polls the email server until it receives the OTP. After identity verification, it goes ahead and completes the rest of the agency profiles and injects a direct file-path to

automate the uploads of documents. This sequential progression makes sure that script proceeds only after every important milestone of this script is confirmed to be met.

```java
// Fetches the latest 6-digit OTP from Gmail
public static String getLatestOTP(String email, String appPassword) throws Exception {
    Properties props = new Properties();
    props.put("mail.store.protocol", "imaps");
    props.put("mail.imaps.host", "imap.gmail.com");
    props.put("mail.imaps.port", "993");
    props.put("mail.imaps.ssl.enable", "true");

    Session session = Session.getInstance(props);
    Store store = session.getStore("imaps");
    store.connect("imap.gmail.com", email, appPassword);

    Folder inbox = store.getFolder("INBOX");
    inbox.open(Folder.READ_ONLY);

    Message[] messages = inbox.search(new FlagTerm(new Flags(Flags.Flag.SEEN), false));

    if (messages.length == 0) {
        inbox.close(false);
        store.close();
        return "";
    }

    Message message = messages[messages.length - 1];
    String content = getTextFromMessage(message);

    Pattern p = Pattern.compile("\\b\\d{6}\\b");
    Matcher m = p.matcher(content);

    String otp = "";
    if (m.find()) {
        otp = m.group(0);
    }

    inbox.close(false);
    store.close();
    return otp;
}
```

5. Script Workflow (Step-by-Step)

- Initialization: Opens Chrome and maximizes the window.

```java
driver.manage().window().maximize();
driver.get("https://authorized-partner.vercel.app/");
```

- Navigation: Leads to the Signup Page out of the Landing Page.

```java
// 1. Navigation
wait.until(ExpectedConditions.elementToBeClickable(By.linkText("Login"))).click();
Thread.sleep(2000);
wait.until(ExpectedConditions.elementToBeClickable(By.linkText("Sign Up"))).click();
```

- Name, Email, Phone, and Password: Data Entry.

```
System.out.println("Filling signup form...");
wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("firstName"))).sendKeys("Mirage");
driver.findElement(By.name("lastName")).sendKeys("Shrestha");
driver.findElement(By.name("email")).sendKeys(email);
driver.findElement(By.name("phoneNumber")).sendKeys("9867078356");
driver.findElement(By.name("password")).sendKeys("Mirage123#");
driver.findElement(By.name("confirmPassword")).sendKeys("Mirage123#");
Thread.sleep(2000);
driver.findElement(By.xpath("//button[text()='Next']")).click();
```

- OTP Polling= The script executes 42 seconds (6 attempts x 7 seconds) to allow the email to be received.

```
System.out.println("Waiting for OTP email...");
String otp = "";
for (int i = 0; i < 6; i++) {
    Thread.sleep(7000);
    otp = getLatestOTP(email, appPassword);
    if (!otp.isEmpty()) break;
    System.out.println("Polling for OTP... attempt " + (i+1));
}

if (otp.isEmpty()) throw new Exception("OTP not found.");
System.out.println("OTP Received: " + otp);
```

- OTP Entry: When located the OTP is inserted into the data-input-otp field.

```
WebElement otpInput = wait.until(ExpectedConditions.elementToBeClickable(
    By.cssSelector("input[data-input-otp='true']")
));
otpInput.click(); // Focus the field
otpInput.sendKeys(otp);
Thread.sleep(3000);
```

- Agency Profile: Accomplishes business-related data (Role, Website, Location).

```
//  7. Agency Details
wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("agency_name"))).sendKeys("Miracle Agency");
driver.findElement(By.name("role_in_agency")).sendKeys("Manager");
driver.findElement(By.name("agency_email")).sendKeys("miracleagency@gmail.com");
driver.findElement(By.name("agency_website")).sendKeys("www.miracleagency.com");
driver.findElement(By.name("agency_address")).sendKeys("Koteshwor, Kathmandu");
driver.findElement(By.cssSelector("button[role='combobox']")).click();
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[contains(text(), 'Australia')]"))).click();
Thread.sleep(2000);
driver.findElement(By.xpath("//button[@type='submit' and text()='Next']")).click();
```

- Professional Metrics: Chooses levels of experience and metrics of recruitment.

```
// 8. Professional Experience
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//button[@role='combobox' and .//span[contains(text(), 'Select Your Experience Level')]]"))).click();
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//div[@role='option' or @role='menuitem']//span[text()='7 years']"))).click();
driver.findElement(By.name("number_of_students_recruited_annually")).sendKeys("500");
driver.findElement(By.name("focus_area")).sendKeys("Undergraduate admissions to Australia");
driver.findElement(By.name("success_metrics")).sendKeys("95");
driver.findElement(By.xpath("//label[contains(.,'Visa Processing')]/preceding-sibling::button")).click();
driver.findElement(By.xpath("//label[contains(.,'Test Prepration')]/preceding-sibling::button")).click();
Thread.sleep(2000);
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//button[text()='Next']"))).click();
```

- File Upload: Adds Company Registration and Achievement Certificates in the local directory.

```
// 9. Verification and Preferences
wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("business_registration_number"))).sendKeys("REG12345678");
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//button[@role='combobox' and .//span[contains(text(), 'Preferred Countries')]]"))).click();
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Australia']"))).click();
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//label[contains(.,'Universities')]/preceding-sibling::button"))).click();
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//label[contains(.,'Colleges')]/preceding-sibling::button"))).click();
driver.findElement(By.name("certification_details")).sendKeys("ICEF Certified Education Agent");
wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("(//input[@type='file'])[1]"))).sendKeys("C:\\Users\\Rabindra Basnet\\Downloads\\Company R
wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("(//input[@type='file'])[2]"))).sendKeys("C:\\Users\\Rabindra Basnet\\Downloads\\High-Scho
Thread.sleep(2000);
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//button[@type='submit' and text()='Submit']"))).click();
```

- Output Console

```
Filling signup form...
Waiting for OTP email...
OTP Received: 219351
Signup Process Completed!
```

6. Execution Requirements.

The environment should be properly configured before the script is run so that the browser and email server can communicate.

- Libraries: You will need to add libraries of Selenium and JavaMail (javax.mail) to your project.
- Gmail Security: Since Gmail prevents the use of standard logins, you are to use an App Password. This you can create in your Google Account security settings.

- File Locations: PDF files as referred in the code should be placed at the precise folder path, otherwise the code will not locate them. You can update them as you want.
- Driver Compatibility: The chromedriver file also has to be in line with the version of the Chrome web browser that runs on your computer.

Conclusion

To sum up, the presented automation script will be a solid and efficient tool to verify the complicated signup and verification framework of the Authorized Partner portal. The script removes the necessity of a manual intervention on the delicate OTP verification step since it manages to bridge the divide between web-based UI interaction and the email protocols on the other end. The resilience properties that it has a built-in capability of retrying elements that turn out stale, explicit synchronization waits, and so on make the test resistant to even changing network performance or quickly loading dynamic content.