

Automate a Web Application(Selenium)

→Login.java

```
package com;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;

public class Login {

    public static void main(String[] args) throws InterruptedException {

        WebDriverManager.chromedriver().setup();

        WebDriver driver = new ChromeDriver();

        // Navigate to the login page

        String path = "login.html";

        String url = Login.class.getClassLoader().getResource(path).toString();

        driver.get(url);

        // Login Automation

        login(driver, "ashok125@gmail.com", "Ashok@123");

        // Pause for 5 seconds (5000 milliseconds) before closing the browser

        Thread.sleep(1000);

        // Close the browser

        driver.quit();

    }
```

```

private static void login(WebDriver driver, String email, String password) throws
InterruptedException {

    WebElement emailInput = driver.findElement(By.id("login-email"));
    WebElement passwordInput = driver.findElement(By.id("login-password"));
    WebElement submitButton = driver.findElement(By.xpath("//button[text()='Submit']"));
    // Fill out the login form
    emailInput.sendKeys(email);
    passwordInput.sendKeys(password);
    Thread.sleep(2000);
    // Click the submit button
    submitButton.click();
    Thread.sleep(1000);
    System.out.println("Login Success");
}
}

```

→Registration.java

```

package com;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;

public class Registration {

    public static void main(String[] args) throws InterruptedException {

        WebDriverManager.chromedriver().setup();

        WebDriver driver = new ChromeDriver();
    }
}

```

```

// Display the welcome page
displayWelcomePage(driver);

// Pause for 2 seconds (2000 milliseconds) to manually complete the registration (for
demonstration purposes)
Thread.sleep(2000);

// Print the "Register Success" message
System.out.println("Register Success");

// Pause for 2 seconds (2000 milliseconds) before displaying the registration page
Thread.sleep(2000);

// Registration Automation
registerNewEmployee(driver, "ashok", "ashok125@gmail.com", "Ashok@123");

// Close the browser
driver.quit();
}

```

```

private static void displayWelcomePage(WebDriver driver) {
    String welcomePath = Registration.class.getResource("/welcome.html").toString();
    driver.get(welcomePath);
}

```

```

private static void registerNewEmployee(WebDriver driver, String name, String email,
String password) throws InterruptedException {
    String registrationPath = Registration.class.getResource("/registration.html").toString();
    driver.get(registrationPath);

    // Fill out the registration form
    WebElement nameInput = driver.findElement(By.id("name"));
    WebElement emailInput = driver.findElement(By.id("email"));
    WebElement passwordInput = driver.findElement(By.id("password"));
    WebElement registerButton = driver.findElement(By.id("register-button"));
}

```

```

nameInput.sendKeys(name);
emailInput.sendKeys(email);
passwordInput.sendKeys(password);
Thread.sleep(2000);

// Click the registration button
registerButton.click();

// Check if the window is still open
if (!driver.getWindowHandles().isEmpty()) {
    // Wait for the success message
    try {
        WebElement successMessage = driver.findElement(By.xpath("//h1[contains(text(),
'Register Success')]"));
        if (successMessage.isDisplayed()) {
            System.out.println("Register Success");
        }
    } catch (org.openqa.selenium.NoSuchElementException e) {
        System.out.println("Registerion page closed");
    }
} else {
    System.out.println("Register Failed: Window closed");
}
}
}

```

→LoginTest.java

```
package com;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
import io.github.bonigarcia.wdm.WebDriverManager;

public class LoginTest {

    private WebDriver driver;

    @BeforeClass

    public void setUp() {

        WebDriverManager.chromedriver().setup();

        driver = new ChromeDriver();

        String path = LoginTest.class.getClassLoader().getResource("login.html").getPath();

        driver.get("file://" + path);

    }

    @Test

    public void testLogin() throws InterruptedException {

        loginEmployee(driver, "ashoksree1611@gmail.com", "Ashok@123");

        // Optionally, add assertions to verify login success or check for error messages

    }

    @AfterClass
```

```

public void tearDown() {
    driver.quit();
}

private void loginEmployee(WebDriver driver, String email, String password) throws
InterruptedException {

    WebElement emailInput = driver.findElement(By.name("email"));
    WebElement passwordInput = driver.findElement(By.name("password"));
    WebElement submitButton = driver.findElement(By.xpath("//button[text()='Submit']"));

    // Fill out the login form
    emailInput.sendKeys(email);
    passwordInput.sendKeys(password);
    Thread.sleep(2000);

    // Click the "Submit" button
    submitButton.click();

    // Optionally, add code to handle login success or failure scenarios
}
}

```

→RegistrationTest.java

```

package com;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

```

```
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
import io.github.bonigarcia.wdm.WebDriverManager;

public class RegistrationTest {

    private WebDriver driver;

    @BeforeClass
    public void setup() {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10, java.util.concurrent.TimeUnit.SECONDS);
    }

    @Test(priority = 1)
    public void testWelcomePage() {
        // Test displaying the welcome page
        displayWelcomePage();

        // Pause for 5 seconds (5000 milliseconds) to manually interact with the page (for
        demonstration purposes)
        sleep(5000);

        // Print the "Welcome Page Test Successful" message
        System.out.println("Welcome Page Test Successful");
    }

    @Test(priority = 2)
    public void testRegistrationProcess() throws InterruptedException {
        // Test the registration process
        displayRegistrationPage();

        // Fill out the registration form
    }
}
```

```

        registerNewEmployee("ashok", "ashok125@gmail.com", "Ashok@123");

        // Pause for 5 seconds (5000 milliseconds) before checking for the "Register Success"
message
        sleep(5000);

        // Check for the "Register Success" message

        boolean successMessageDisplayed =
isElementDisplayed(By.xpath("//h1[contains(text(), 'Register Success')]"));

        if (successMessageDisplayed) {

            System.out.println("Register Success");

        }

    }
}

```

@AfterClass

```

public void teardown() {

    // Close the browser after the test is executed

    if (driver != null) {

        driver.quit();

    }

}

```

```

private void displayWelcomePage() {

    String welcomePath = RegistrationTest.class.getResource("/welcome.html").toString();

    driver.get(welcomePath);

}

```

```

private void displayRegistrationPage() {

    String registrationPath =
RegistrationTest.class.getResource("/registration.html").toString();

    driver.get(registrationPath);

}

```



```
private void registerNewEmployee(String name, String email, String password) throws  
InterruptedException {
```

```
    WebElement nameInput = driver.findElement(By.id("name"));
```

```
    WebElement emailInput = driver.findElement(By.id("email"));
```

```
    WebElement passwordInput = driver.findElement(By.id("password"));
```

```
    WebElement registerButton = driver.findElement(By.id("register-button"));
```

```
    nameInput.sendKeys(name);
```

```
    emailInput.sendKeys(email);
```

```
    passwordInput.sendKeys(password);
```

```
    Thread.sleep(2000);
```

```
    // Click the registration button
```

```
    registerButton.click();
```

```
}
```

```
private boolean isElementDisplayed(By locator) {
```

```
    try {
```

```
        return driver.findElement(locator).isDisplayed();
```

```
    } catch (org.openqa.selenium.NoSuchElementException e) {
```

```
        return false;
```

```
    }
```

```
}
```

```
private void sleep(long milliseconds) {
```

```
    try {
```

```
        Thread.sleep(milliseconds);
```

```
    } catch (InterruptedException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
}
```

→Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>

</head>
<body>
  <h1>Login Page</h1>
  <form id="login-form">
    <label for="login-email">Email:</label>
    <input type="email" id="login-email" name="email" required><br>

    <label for="login-password">Password:</label>
    <input type="password" id="login-password" name="password"
required><br>

    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

→Registration.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration Page</title>

</head>
<body>
  <h1>Registration</h1>
  <form id="registration-form" action="/register" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>
    <br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
    <br>
    <label for="password">Password:</label>
```

```

        <input type="password" id="password" name="password" required>
        <br>
        <input type="submit" value="Register" id="register-button">
    </form>
</body>
</html>

```

→Welcome.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome Page</title>

</head>
<body>
    <h1>Welcome to Our Website!</h1>

    <p>Click the button below to proceed to the registration page:</p>
    <a href="registration.html">Registration Page</a>
</body>
</html>

```

→Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>login</groupId>
    <artifactId>login</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>4.10.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-support -->
        <dependency>

```

```

    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-support</artifactId>
    <version>4.10.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-chrome-driver</artifactId>
    <version>4.10.0</version>
</dependency>
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>5.2.2</version> <!-- Use the latest version available -->
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-simple</artifactId>
        <version>1.7.32</version>
    </dependency>
</dependencies>
</project>

```

→ Default test.html

```

<html>
<head>
<title>TestNG: Default test</title>
<link href="../testng.css" rel="stylesheet" type="text/css" />
<link href="../my-testng.css" rel="stylesheet" type="text/css" />

<style type="text/css">
.log { display: none;}
.stack-trace { display: none;}
</style>
<script type="text/javascript">
<!--
function flip(e) {
    current = e.style.display;
    if (current == 'block') {
        e.style.display = 'none';
    }
}

```

```

        return 0;
    }
    else {
        e.style.display = 'block';
        return 1;
    }
}

function toggleBox(szDivId, elem, msg1, msg2)
{
    var res = -1; if (document.getElementById) {
        res = flip(document.getElementById(szDivId));
    }
    else if (document.all) {
        // this is the way old msie versions work
        res = flip(document.all[szDivId]);
    }
    if(elem) {
        if(res == 0) elem.innerHTML = msg1; else elem.innerHTML = msg2;
    }

}

function toggleAllBoxes() {
    if (document.getElementsByTagName) {
        d = document.getElementsByTagName('div');
        for (i = 0; i < d.length; i++) {
            if (d[i].className == 'log') {
                flip(d[i]);
            }
        }
    }
}

// -->
</script>

</head>
<body>
<h2 align='center'>Default test</h2><table border='1' align="center">
<tr>
<td>Tests passed/Failed/Skipped:</td><td>1/0/0</td>
</tr><tr>
<td>Started on:</td><td>Thu Jul 20 11:21:54 IST 2023</td>
</tr>
<tr><td>Total time:</td><td>4 seconds (4374 ms)</td>
</tr><tr>
<td>Included groups:</td><td></td>

```

```

</tr><tr>
<td>Excluded groups:</td><td></td>
</tr>
</table><p/>
<small><i>(Hover the method name to see the test class name)</i></small><p/>
<table width='100%' border='1' class='invocation-passed'>
<tr><td colspan='4' align='center'><b>PASSED TESTS</b></td></tr>
<tr><td><b>Test method</b></td>
<td><b>Attribute(s)</b></td>
<td width="30%"><b>Exception</b></td>
<td width="10%"><b>Time (seconds)</b></td>
<td><b>Instance</b></td>
</tr>
<tr>
<td title='com.LoginTest.testLogin()'><b>testLogin</b><br>Test class:
com.LoginTest</td>
<td></td>
<td>2</td>
<td>com.LoginTest@1f2586d6</td></tr>
</table><p>
</body>
</html>

```

→Testng-failed.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Failed suite [Default suite]" guice-stage="DEVELOPMENT">
  <test thread-count="5" name="Default test(failed)">
    <classes>
      <class name="com.RegistrationTest">
        <methods>
          <include name="setup"/>
          <include name="teardown"/>
          <include name="testRegistrationProcess"/>
        </methods>
      </class> <!-- com.RegistrationTest -->
    </classes>
  </test> <!-- Default test(failed) -->
</suite> <!-- Failed suite [Default suite] -->

```

→Default test.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by org.testng.reporters.JUnitXMLReporter -->
<testsuite ignored="0" hostname="maneesh" failures="0" tests="1" name="Default
test" time="4.374" errors="0" timestamp="2023-07-20T11:21:59 IST">
  <testcase classname="com.LoginTest" name="testLogin" time="2.244"/>
</testsuite> <!-- Default test -->
```

→Classes.html

```
<table border='1'>
<tr>
<th>Class name</th>
<th>Method name</th>
<th>Groups</th>
</tr><tr>
<td>com.LoginTest</td>
<td>&nbsp;</td><td>&nbsp;</td></tr>
<tr>
<td align='center' colspan='3'>@Test</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>testLogin</td>
<td>&nbsp;</td></tr>
<tr>
<td align='center' colspan='3'>@BeforeClass</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>setUp</td>
<td>&nbsp;</td></tr>
<tr>
<td align='center' colspan='3'>@BeforeMethod</td>
</tr>
<tr>
<td align='center' colspan='3'>@AfterMethod</td>
</tr>
<tr>
<td align='center' colspan='3'>@AfterClass</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>tearDown</td>
<td>&nbsp;</td></tr>
</table>
```

