

Selenium Automation Testing Project Documentation - S Marshall (52287740)

1. Project Overview

This document provides a structured guide for handling common exceptions and their solutions in a Selenium automation testing project. It includes project structure, root cause analysis, fixes, and best practices.

2. Project Structure

```
src/test/java/pages
    Homepage.java
    Login.java
    Registration.java
    Testcase.java
src/test/java/tests
    TestingClass.java
src/test/java/utilities
    CommonMethods.java
    configreader.java
```

3. Exception Handling

[Homepage.java](#)

Package: src/test/java/pages

Possible Exceptions:

- NullPointerException: Occurs if dr is null or PageFactory.initElements() not executed correctly.
- ElementNotInteractableException: Happens when register element is not visible or clickable.
- FileNotFoundException: Incorrect path for log4j.properties.
- Exception(s):
 - NullPointerException

- If dr (WebDriver) is not initialized before calling Login constructor.
- NoSuchElementException
- If locators like loginLink, emailField, or passwordField are incorrect or element is not present in DOM.
- TimeoutException
- If accountLink does not appear within the wait duration in verify() method.
- FileNotFoundException
- If log4j.properties path is incorrect in PropertyConfigurator.configure().
- ElementNotInteractableException
- If loginLink or logoutLink is hidden or disabled when clicked.

Root Cause Analysis:

Fix Steps:

Ensure WebDriver is initialized before calling Homepage constructor:

```
WebDriver driver = new ChromeDriver();

Homepage home = new Homepage(driver);

Use explicit wait before clicking:
wait.until(ExpectedConditions.elementToBeClickable(register)).click();

Correct log4j path:
PropertyConfigurator.configure("src/test/resources/log4j.properties");

p2 - Describe
```

Login.java

Package: src/test/java/pages

Possible Exceptions:

- WebDriver not initialized → NullPointerException.
- Incorrect XPath or ID → NoSuchElementException.
- Element not loaded → TimeoutException.
- Wrong resource path → FileNotFoundException.
- Element hidden → ElementNotInteractableException.
- Exception(s):

- NullPointerException
- If dr (WebDriver) is not initialized before calling the constructor.
- NoSuchElementException
 - If locators like FirstName, LastName, or register-button are incorrect or element is missing in DOM.
- TimeoutException
 - If register button does not become clickable within the wait duration.
- ElementNotInteractableException
 - If radio button or register button is hidden or disabled.
- FileNotFoundException
 - If log4j.properties path is incorrect in PropertyConfigurator.configure().

Root Cause Analysis:

Fix Steps:

Initialize WebDriver before creating Login object:

```
WebDriver driver = new ChromeDriver();
Login loginPage = new Login(driver);
```

Use explicit waits for clickable elements:

```
wait.until(ExpectedConditions.elementToBeClickable(loginLink)).click();
```

Correct log4j path:

```
PropertyConfigurator.configure("src/test/resources/log4j.properties");
```

Validate locators using browser dev tools before coding.

p3 - Describe

Registration.java

Package: src/test/java/pages

Possible Exceptions:

- WebDriver not initialized → NullPointerException.
- Incorrect XPath or ID → NoSuchElementException.
- Element not loaded → TimeoutException.

- Wrong resource path → FileNotFoundException.
- Element hidden → ElementNotInteractableException.
- Exception(s):
 - NullPointerException
 - If dr (WebDriver) is not initialized before calling the constructor.
 - TimeoutException
 - If computerText, electronicsText, or apparelText elements do not appear within the wait duration.
 - NoSuchElementException
 - If locators for header menu links are incorrect or elements are missing in DOM.
 - FileNotFoundException
 - If log4j.properties path is incorrect in PropertyConfigurator.configure().
 - ElementNotInteractableException
 - If elements are hidden or disabled when checked for visibility.

Root Cause Analysis:

Fix Steps:

Initialize WebDriver before creating Registration object:

```
WebDriver driver = new ChromeDriver();
Registration regPage = new Registration(driver);
```

Use explicit waits for clickable elements:

```
wait.until(ExpectedConditions.elementToBeClickable(register)).click();
```

Correct log4j path:

```
PropertyConfigurator.configure("src/test/resources/log4j.properties");
```

Validate locators using browser dev tools before coding.

p4 - Describe

Testcase.java

Package: src/test/java/pages

Possible Exceptions:

- WebDriver not initialized → NullPointerException.
- Incorrect XPath or missing elements → NoSuchElementException.
- Element not loaded → TimeoutException.
- Wrong resource path → FileNotFoundException.
- Element hidden → ElementNotInteractableException.
- .ignoring(NoSuchElementException.class);
- Correct log4j path:
 - PropertyConfigurator.configure("src/test/resources/log4j.properties");
 - Validate locators using browser dev tools before coding.
- p5 – Describe
 - Exception(s):
 - NullPointerException
 - If dr (WebDriver) is not initialized before calling page object constructors.
 - FileNotFoundException
 - If log4j.properties path is incorrect in PropertyConfigurator.configure().
 - IllegalArgumentException
 - If @Parameters("BROWSER") value is invalid or not passed from parametersuite.xml.
 - NoSuchElementException / TimeoutException
 - If elements in Homepage, Registration, Login, or Testcase pages are not found during test execution.
 - AssertionError
 - If Assert.assertTrue() fails in login or text verification tests.
 - IOException
 - If configreader fails to read config.properties.

Root Cause Analysis:

Fix Steps:

Initialize WebDriver before creating Testcase object:

```
WebDriver driver = new ChromeDriver();
```

```
Testcase tcPage = new Testcase(driver);
```

Use explicit waits for visibility:

```
wait.until(ExpectedConditions.visibilityOf(computerText));
```

Show more lines

For dynamic waits, use FluentWait (already implemented for electronicsText):

```
Wait<WebDriver> fluentWait = new FluentWait<>(driver)
```

```
.withTimeout(Duration.ofSeconds(10))
```

```
.pollingEvery(Duration.ofMillis(500))
```

TestingClass.java

Package: src/test/java/tests

Possible Exceptions:

- Driver not launched → NullPointerException.
- Incorrect resource path → FileNotFoundException.
- Missing or wrong browser parameter → IllegalArgumentException.
- Incorrect locators or slow page load → NoSuchElementException / TimeoutException.
- Wrong expected text → AssertionError.
- Missing config file → IOException.
- Exception(s):
 - NullPointerException
 - If configreader fails to load url or dr is not initialized properly.
 - IllegalArgumentException
 - If browser parameter passed to launchBrowser() is invalid.
 - FileNotFoundException / IOException

- If config.properties file is missing or unreadable.
- SessionNotCreatedException
- If ChromeDriver or EdgeDriver executable version is incompatible with the browser.
- WebDriverException
- If driver path is incorrect or browser fails to launch.

Root Cause Analysis:

Fix Steps:

Ensure browser parameter is passed correctly in parametersuite.xml:

```
<parameter name="BROWSER" value="chrome"/>
```

Launch browser before initializing page objects:

```
launchBrowser(browser);

ho = new Homepage(dr);

reg = new Registration(dr);

login = new Login(dr);

test = new Testcase(dr);
```

Correct log4j path:

```
PropertyConfigurator.configure("src/test/resources/log4j.properties");
```

Validate locators and use explicit waits in page classes.

Ensure config.properties exists and keys match:

```
String firstname = config.getProperty("first");
```

Add assertions with meaningful messages:

```
Assert.assertTrue(isLoggedIn, "Login failed or account text mismatch");
```

p6 - Describe

CommonMethods.java

Package: src/test/java/utilities

Possible Exceptions:

- Missing or incorrect driver path → WebDriverException.
- Wrong browser name → IllegalArgumentException.

- Missing config file or key → FileNotFoundException / IOException.
- Driver version mismatch → SessionNotCreatedException.
- dr not initialized → NullPointerException.
- Exception(s):
 - FileNotFoundException
 - If config.properties file is missing or the path is incorrect.
 - IOException
 - If there is an error while loading properties from the file.
 - NullPointerException
 - If prop is not initialized properly or the requested key does not exist.

Root Cause Analysis:

Fix Steps:

Validate browser parameter before launching:

```
if (browser.equalsIgnoreCase("chrome")) {
    System.setProperty("webdriver.chrome.driver", "chromedriver_v141.exe");
    dr = new ChromeDriver();
} else if (browser.equalsIgnoreCase("edge")) {
    System.setProperty("webdriver.edge.driver", "edgedriver_v141.exe");
    dr = new EdgeDriver();
}
```

Ensure config.properties exists and contains url key:

```
String url = config.getProperty("url");
```

Correct driver executable paths if manually setting:

```
System.setProperty("webdriver.chrome.driver",
"path/to/chromedriver.exe");
```

p7 - Describe

configreader.java

Package: src/test/java/utilities

Possible Exceptions:

- Incorrect file path → FileNotFoundException.
- Error during file reading → IOException.
- Missing key or uninitialized prop → NullPointerException.
- throw new FileNotFoundException("Config file not found at: " + file.getAbsolutePath());
- }
- Handle exceptions gracefully:
 - try (FileInputStream fis = new FileInputStream("src/test/resources/config.properties")) {
 - prop = new Properties();
 - prop.load(fis);
 - } catch (IOException e) {
 - e.printStackTrace();
 - }
- Check for null keys before returning:
 - String value = prop.getProperty(key);
 - if (value == null) {
 - throw new IllegalArgumentException("Key not found: " + key);
 - }
- return value;

Root Cause Analysis:

Fix Steps:

Ensure correct file path for config.properties:

```
FileInputStream fis = new  
FileInputStream("src/test/resources/config.properties");
```

Validate file existence before loading:

```
File file = new File("src/test/resources/config.properties");  
if (!file.exists()) {
```

4. Best Practices

- Always initialize WebDriver before creating page objects.
- Validate locators using browser developer tools.
- Use explicit waits for dynamic elements.
- Maintain correct resource paths for properties and drivers.
- Handle exceptions gracefully with meaningful messages.
- Ensure config.properties file exists and keys match.
- Add assertions with clear messages for better debugging.