Sikuli for UI Testing

This section will guide you to:

- Integrate Sikuli with Selenium WebDriver and interact with web elements

Development Environment:

- Eclipse IDE for Enterprise Java Developers Version Oxygen.3a Release (4.7.3a)
- Java Development Kit Version 8

This guide has mainly four subsections, namely:

4.11.1 Integrating Sikuli with Selenium WebDriver

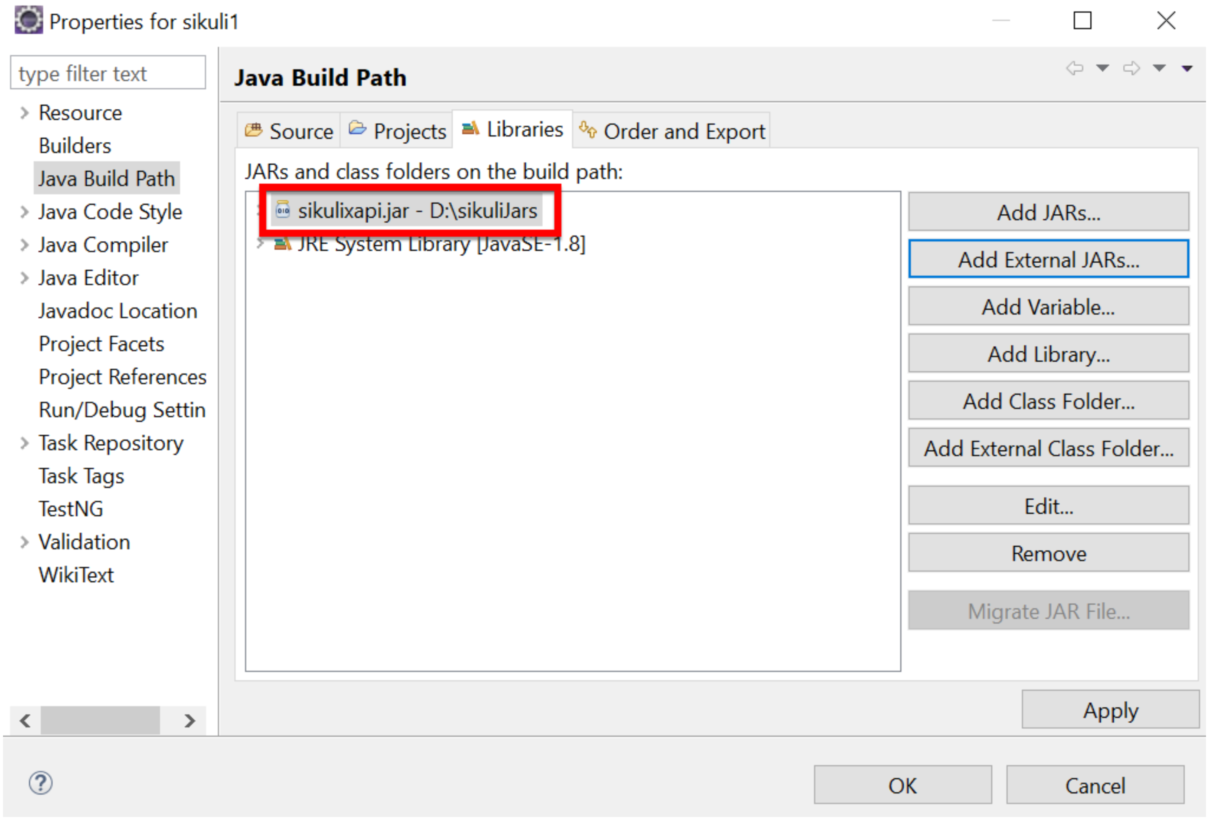
4.11.2 Screen class in Sikuli

4.11.3 Pattern class in Sikuli

4.11.4 Pushing the code to your GitHub repositories

Steps 4.11.1: Integrating Sikuli with Selenium WebDriver

- Sikuli Jar files are already present in your practice labs. To learn about its directory path details, you can refer to the lab guide for Phase 1.
- Open Eclipse and create a new Java project
- Right-click on the project. Navigate through the given path: Build path->Configure build path->Add external Jars.
- Click on **Apply and OK.**



JARs and class folders on the build path:

- sikulixapi.jar - D:\sikuliJars
- JRE System Library [JavaSE-1.8]

Steps 4.11.2: Screen class in Sikuli

- Screen class is a base class which contains some predefined methods to perform operations, such as click, double click, providing input to the text box and hover, etc.

- Below are the commonly used methods:

  - Click

  Syntax: Screen s = new Screen();
  s.click()("imag.png”);

  - doubleClick

  Syntax: Screen s = new Screen();
  s.doubleClick()("imag.png”);

  - Type

  Syntax: s.type(“imag.png”, “Text”);

  - Hover

  Syntax: s.hover(“imag.png”);

  - Find

Syntax: s.find(“imag.png”);

Steps 4.11.3: Pattern class in Sikuli

- Pattern class is used to associate the image file to identify the element
- Pattern class takes the path of the image as a parameter
- Below are the commonly used methods:

    - getFileName

    Syntax: Pattern p = new Pattern(“D:\Test\imag.png”)

    - Similar

    Syntax: Pattern p1 = p.similar Pattern(“0.7f”);

    - Exact

    Syntax: Pattern p1 = p.exact();

The script looks like this:

```
package sikuli1;
import org.sikuli.script.FindFailed;
import org.sikuli.script.Pattern;
import org.sikuli.script.Screen;

public class SikuliClass {
        public static void main(String[] args ) throws FindFailed {
                Screen s = new Screen();
                Pattern p = new
Pattern("C:\\Users\\Testing\\Desktop\\siluli\\Capture.PNG");
                s.doubleClick(p);
        }
}
```

- Run the script and notice the action performed on the image (The path, which we have mentioned in the script).

Steps 4.11.4: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

```
git add .
```

- Commit the changes using the following command:

```
git commit .  -m “Changes have been committed.”
```

- Push the files to the folder you initially created using the following command:

```
git push -u origin master
```