Mobile automation testing

- Android Automation
- IOS automation

Android Automation

Install node Js.

- Install Node JS .Refer to this link - https://nodejs.org/en/download/prebuilt-installer.
- Use the **node -v** command to check the installed version of Node.js.
- Use the **npm -v** command to check the version of npm.

Appium

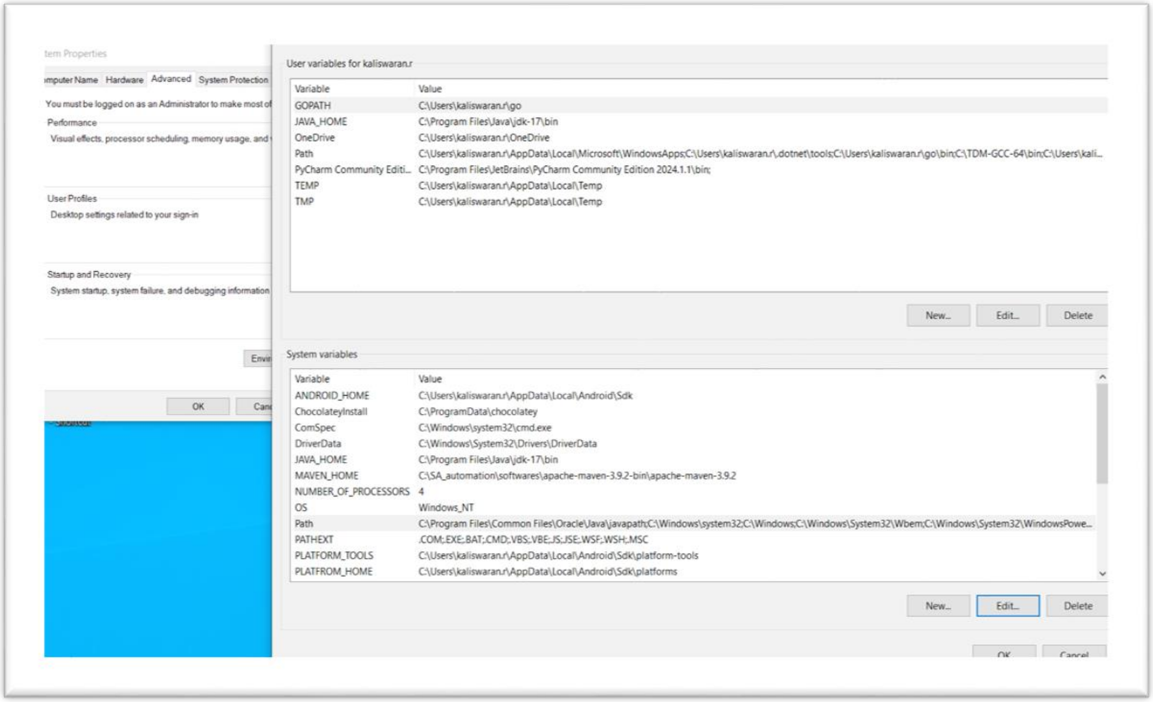
- **npm install -g appium** . This commend some times did not work properly, at the time we are use **sudo npm install –g appium** use this command.
- **npm install -g appium** .This commed is used to install appium.
- This command will install Appium globally on your system.
- Use the **appium --version** command to check the globally installed version of Appium.

Appium driver uiautomator2

- Use this **appium driver install uiautomator2** commend to install uiautomator2 driver

Android Studio

- Link for download android studio - https://developer.android.com/studio?gad_source=1&gclid=CjwKCAjw9cCyBhBzEiwAJTUWNehjcouLjzxFlGLvONJgdFuVaq5PwPMC7p_h7PsOXJ3xvZ3NLMRaXBoC9A4QAvD_BwE&gclsrc=aw.ds
- **Set environment variables** –Android sdk, Android home/tools, Android home /platform-tools .



tem Properties
mputer Name
Hardware
Advanced
System Protection
You must be logged on as an Administrator to make most of
Performance
Visual effects, processor scheduling, memory usage, and
User Profiles
Desktop settings related to your sign-in
Startup and Recovery
System startup, system failure, and debugging information
Envi
OK
Canc
New...
Edit...
Delete
New...
Edit...
Delete
OK
Cancel

User variables for kaliswaran.r

Variable	Value
GOPATH	C:\Users\kaliswaran.r\go
JAVA_HOME	C:\Program Files\Java\jdk-17\bin
OneDrive	C:\Users\kaliswaran.r\OneDrive
Path	C:\Users\kaliswaran.r\AppData\Local\Microsoft\WindowsApps;C:\Users\kaliswaran.r\.dotnet\tools;C:\Users\kaliswaran.r\go\bin;C:\TDM-GCC-64\bin;C:\Users\kali...
PyCharm Community Editi...	C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.1\bin;
TEMP	C:\Users\kaliswaran.r\AppData\Local\Temp
TMP	C:\Users\kaliswaran.r\AppData\Local\Temp

System variables

Variable	Value
ANDROID_HOME	C:\Users\kaliswaran.r\AppData\Local\Android\Sdk
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk-17\bin
MAVEN_HOME	C:\SA_automation\softwares\apache-maven-3.9.2-bin\apache-maven-3.9.2
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowe...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PLATFORM_TOOLS	C:\Users\kaliswaran.r\AppData\Local\Android\Sdk\platform-tools
PLATFROM_HOME	C:\Users\kaliswaran.r\AppData\Local\Android\Sdk\platforms

Appium-doctor

- **npm install -g appium-doctor** command through the Command Prompt, you can use the **appium-doctor** command to check the setup of Appium.

Get UDID

- Enable **devloper mode** and **USB debugging** option.
- Now, connect the phone to the system via USB, and execute the **adb devices** command to retrieve the UDID for the connected devices..

Appium inspector

- Link for download - **Appium-Inspector-2024.3.4-win-x64.exe.**
- After launching Appium Inspector, you need to start the Appium server.
- To start the Appium server, execute the command **appium**.
- Connection between the Inspector and the mobile device.

Desired capabilities

```
{
“platformName”: “Android”,
“platformVersion”: “10.0”,
“deviceName”: “Samsung Galaxy Note10”,
“automationName”: “Appium”,
“app”: “path for the app under test”
}
```

➢ Prerequisites for android automation

Java Development Kit (JDK):

- Link - https://www.oracle.com/in/java/technologies/downloads/
- After downloading the Java JDK, set the environment variable on your system.

Eclipse : Download latest version - eclipse-inst-jre-win64.exe

➢ Get AppPackage, appActivity

- Use the command **dumpsys window displays | grep -E “mCurrentFocus”** to get the current application details.

➢ Maven dependencies

Required dependencies to launch the application.

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java >

<dependency>

   <groupId>org.seleniumhq.selenium</groupId>

   <artifactId>selenium-java</artifactId>

   <version>4.21.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

   <groupId>org.testng</groupId>

   <artifactId>testng</artifactId>

   <version>7.10.2</version>

   <scope>test</scope>

</dependency>


<!-- https://mvnrepository.com/artifact/io.appium/java-client -->
```

```
<dependency>
  <groupId>io.appium</groupId>
  <artifactId>java-client</artifactId>
  <version>9.2.2</version>
</dependency>
```

- **.Java code**
- **Sample code to launch the application.**

```
package pac
import java.net.MalformedURLException;
import java.net.URL;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.remote.DesiredCapabilities;
import io.appium.java_client.android.AndroidDriver;
public class Launch_App {
public static AndroidDriver driver;
public static void main(String[] args) throws MalformedURLException {
 DesiredCapabilities cap=new DesiredCapabilities();
cap.setCapability("platformName", "Android");
 cap.setCapability("automationName", "UiAutomator2");
 cap.setCapability("platformVersion","12.0");
 cap.setCapability("udid","YHB6TOS4Q8W8PF9X");
 cap.setCapability("deviceName","vivo 1915");
 cap.setCapability("appPackage", "com.android.bbkcalculator");
 cap.setCapability("appActivity", "com.android.bbkcalculator.Calculator");
 URL url=new URL("http://localhost:4723/");
  driver=new AndroidDriver(url,cap);
 }
```

}

❖ IOS Automation

➢ Install node Js.

- Install Node JS .Refer to this link - https://nodejs.org/en/download/prebuilt-installer.
- Use the **node -v** command to check the installed version of Node.js.
- Use the **npm -v** command to check the version of npm.

➢ Appium

- **npm install -g appium** . This commend some times did not work properly, at the time we are use **sudo npm install –g appium** use this command.
- This command will install Appium globally on your system.
- Use the **appium --version** command to check the globally installed version of Appium.

➢ XCUITest

- Use the **appium driver install xcuitest** command to install xcuitest driver.
- **appium –version ,**This command will display the version of Appium installed on your system, including the version of the XCUITest driver.

➢ Xcode

- Download Xcode through the App Store.
- When you install Appium, the WebDriverAgent.xcodeproj project is created under Appium.
- Once have opened the WebDriverAgent.xcodeproj project in Xcode.
- Click on the WebDriverAgent -> WebDriverLib -> set the unique bundle ID, and log in with devloper Apple ID and change the unique bundle id.
- Then, click on the WebDriverAgent -> WebDriverRunner -> set the same bundle ID, and log in with your Apple ID.
- After completing these setups, click on the "Product" menu and then select "Test" to initiate the testing process.
- This will clone the WebDriverAgent onto the connected mobile.

➢ Appium-doctor

- **npm install -g appium-doctor** command through the Command Prompt, you can use the **appium-doctor** command to check the setup of Appium.

➢ Get UDID

- Enable **devloper mode** and **USB debugging** option.

- Unlock your iPhone.
- Go to the top menu and click on "Window" > "Devices and Simulators".
- Display the connected device UDID.

Appium inspector

- Link for download - **Appium-Inspector-2024.3.4-mac-arm64.zip**
- After launching Appium Inspector, you need to start the Appium server.
- To start the Appium server, execute the command 'appium'.

Bundel ID

- For a comprehensive reference on bundle IDs in iOS, you can visit the following link: Apple's official documentation on bundle IDs

Desired capabilities

- Ex:

```
{ "platformName": "iOS",
"platformVersion": "11.0",
"deviceName": "iPhone 7",
"automationName": "XCUITest",
"app": "/path/to/my.app" }
```

Prerequisites for ios automation

Java Development Kit (JDK): Ensure JDK is installed on your machine.

- Link - https://www.oracle.com/in/java/technologies/downloads/
- After downloading the Java JDK, set the environment variable on your system.

Eclipse : Download latest version - eclipse-inst-jre-win64.exe

Maven dependencies

Required dependencies to launch the application.

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java >

<dependency>

   <groupId>org.seleniumhq.selenium</groupId>

   <artifactId>selenium-java</artifactId>

   <version>4.21.0</version>

</dependency>
```

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
   <groupId>org.testng</groupId>
   <artifactId>testng</artifactId>
   <version>7.10.2</version>
   <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/io.appium/java-client -->
<dependency>
   <groupId>io.appium</groupId>
   <artifactId>java-client</artifactId>
   <version>9.2.2</version>
</dependency>
```

- **.Java code**
- **Sample code to launch the application.**

```
package ios_demo;
import java.net.MalformedURLException;
import java.net.URL;

import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.remote.DesiredCapabilities;
import io.appium.java_client.ios.IOSDriver;
public class IOS_Testing_LaunchCalculator {
public static IOSDriver driver;
public static void main(String[] args) throws MalformedURLException {
     DesiredCapabilities cap=new DesiredCapabilities();
     cap.setCapability("platformName","iOS");
     cap.setCapability("automationName","XCUITest");
     cap.setCapability("udid","f5bb7ad592d34531f740769c90a08b0fc1cded54");
    cap.setCapability("deviceName","phone1");
    URL url=new ~~URL~~("http://localhost:4723/");
     cap.setCapability("bundleId","com.apple.calculator");
    driver=new IOSDriver(url,cap);
    IOS_Testing_LaunchCalculator obj=new IOS_Testing_LaunchCalculator();
    obj.Addition();
    System.out.println("Successfully add the value");
     obj.Subtraction();
      System.out.println("Successfully sub the value");
    }
}
```