Requirements for mobile testing:

www.qaautomated.com/2016/01/setting-up-appium-with-android-studio.html

1.Android studio

2.appium jar files for java

3.latest Appium Clinet library

4.appium server

5.Install java.

Android studio installation:

1.Go to <https://developer.android.com/studio/install.html>

Appium jar files for java installation:

1.go to <https://appium.io/downloads.html>

2.click on “java” link under Appium client Libraries.

latest Appium Client library:

1.go to <https://docs.seleniumhq.org/download/>

### 2.click on download for java under Selenium Client & WebDriver Language Bindings

### Install Appium server :

### 1.Go to <https://appium.io/>

### 2.download Appium download button

### Install java

### Set up the environment variable s for java and android.

### Environmental variables->under uservariables section

### Click new and provide name as ANDROD\_HOME and enter value as c:\users\10555\AppData\Local\Android\sdk

### In system variable section

### New -> name as JAVA\_HOME and value as c:\programfiles\java\jdk\_1.8.0\_111

### Open android studio and click on start a new android Studio project.

### Enter application name and click on next button.

### Phone and Tablet minimum version

### Select blank Activity-> next->enter Activity Name ->finish

### Project-> expand app-> click on android and select project from the dropdown

### 

### 

### 

### 

### 

### Here the lib folder is empty.

### 

### Select all jar files and right click and add to library -> create library click on Ok button.

### select build.gradle file.

### Build-> Click on rebuild project

### 

### 

### <https://www.youtube.com/channel/UCjYfNeA94isOHEYQqJNggqw>

### writing testcases:

### we can write the testcases under test package.

### 

### Wite code as below

import java.net.MalformedURLException;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.CapabilityType;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class WaitTests {

WebDriver driver;

@Before

public void setUp() throws MalformedURLException {

// Created object of DesiredCapabilities class.

DesiredCapabilities capabilities = new DesiredCapabilities();

// Set android deviceName desired capability. Set your device name.

capabilities.setCapability("deviceName", "XT1562");

// Set BROWSER\_NAME desired capability. It's Android in our case here.

capabilities.setCapability(CapabilityType.BROWSER\_NAME, "Android");

// Set android VERSION desired capability. Set your mobile device's OS version.

capabilities.setCapability(CapabilityType.VERSION, "6.0.1");

// Set android platformName desired capability. It's Android in our case here.

capabilities.setCapability("platformName", "Android");

// Set android appPackage desired capability. It is

// com.android.calculator2 for calculator application.

// Set your application's appPackage if you are using any other app.

capabilities.setCapability("appPackage", "com.android.calculator2");

// Set android appActivity desired capability. It is

// com.android.calculator2.Calculator for calculator application.

// Set your application's appPackage if you are using any other app.

capabilities.setCapability("appActivity", "com.android.calculator2.Calculator");

// Created object of RemoteWebDriver will all set capabilities.

// Set appium server address and port number in URL string.

// It will launch calculator app in android device.

driver = new RemoteWebDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

}

@Test

public void testFirstCalculator() {

// Click on DELETE/CLR button to clear result text box before running test.

driver.findElements(By.xpath("//android.widget.Button")).get(0).click();

// Click on number 2 button.

driver.findElement(By.name("7")).click();

driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);

}

@After

public void End() {

driver.quit();

}

}

### Double click on Appium server to start the Appium.and click on android icon.

### 

### Click on settings

### 

### Change the port number same as android studio.

### 

### Click on start button for Appium

### 

### Go to your mobile->settings->developer options

### 

### To enable the developer option click on about phone go to build number and click 7 times then the developer option is enabled.

### 

### To run the program

### Select classman and ->rc and select run first test.

### 

### Take xpath from the UI automater

### Xpath syntax:

### //tagName[@attribute=’value’]

### 

### Here tagname is class ie android.widget.Textview

### //tagName[@attribute=’value’]

### Dr.findelementBy.xpath(“//android.widget.Textview[@text=’preference’]”);