**MOBILE GESTURES**

*[Please note: For tap(), withElement(element(WebElement)), longPress(), Duration() -> you must have to manually import all the packages]*

* //This packagae need to import manually for tapOptions();

**import** **static** io.appium.java\_client.touch.TapOptions.*tapOptions*;

* //This package needs to be imported manually for withElement(element(expandleElement))-->element;

**import** **static** io.appium.java\_client.touch.offset.ElementOption.*element*;

* //This package needs to be imported manually for longPressOptions option;

**import** **static** io.appium.java\_client.touch.LongPressOptions.*longPressOptions*;

* //This package needs to be imported manually for ofSeconds() option;

**import** **static** java.time.Duration.*ofSeconds*;

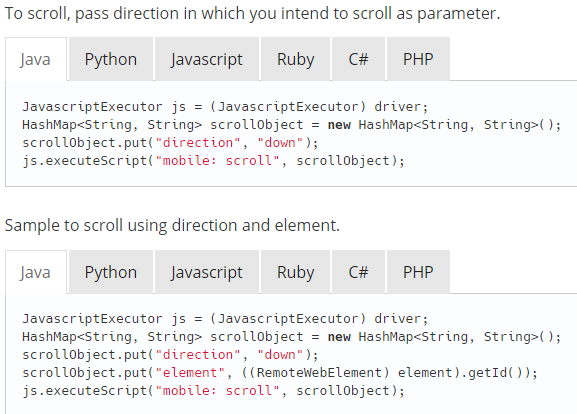
----------------------------------------------------------------------------

* What are the touch objects?

The available events from the spec are: \* press \* release \* moveTo \* tap \* wait \* longPress \* cancel \* perform.

* **Scrolling**

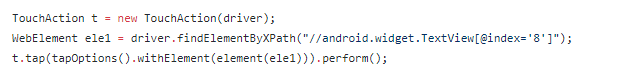
To allow access to this special feature, we override the execute or executeScript methods in the driver, and prefix the command with mobile:. See examples below:



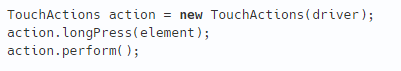
* **Scrolling on a particular element on the screen**:

driver.findElementByAndroidUIAutomator("new UiScrollable(new UiSelector().scrollable(true).instance(0)).scrollIntoView(new UiSelector().textContains(\"WebView\").instance(0))").click();

* TAP:



* LONG PRESS:



C:\Users\User\Desktop\1.PNG

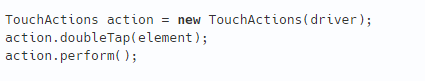
* SWIPE:

C:\Users\User\Desktop\1.PNG

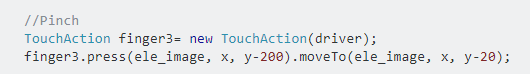
* DRAG AND DROP”

C:\Users\User\Desktop\1.PNG

* DOUBLE TAPP:



* PINCH:



//appium server details

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

//wait

Thread.sleep(4000);

driver.findElementById("com.davemorrissey.labs.subscaleview.sample:id/basicFeatures").click();

Thread.sleep(4000);

WebElement ele\_image = driver.findElementById("com.davemorrissey.labs.subscaleview.sample:id/imageView");

int x1= ele\_image.getLocation().getX();

int y1= ele\_image.getLocation().getY();

System.out.println("x is "+x1+"y1 is "+y1);

int x=ele\_image.getLocation().getX()+ele\_image.getSize().getWidth()/2;

int y= ele\_image.getLocation().getY()+ele\_image.getSize().getHeight()/2;

//Zoom

TouchAction finger1= new TouchAction(driver);

finger1.press(ele\_image, x, y-20).moveTo(ele\_image, x, y-200);

TouchAction finger2= new TouchAction(driver);

finger2.press(ele\_image, x, y+20).moveTo(ele\_image, x, y+200);

MultiTouchAction action= new MultiTouchAction(driver);

action.add(finger1).add(finger2).perform();

Thread.sleep(8000);

//Pinch

TouchAction finger3= new TouchAction(driver);

finger3.press(ele\_image, x, y-200).moveTo(ele\_image, x, y-20);

TouchAction finger4= new TouchAction(driver);

finger4.press(ele\_image, x, y+200).moveTo(ele\_image, x, y+20);

MultiTouchAction action2= new MultiTouchAction(driver);

action2.add(finger3).add(finger4).perform();

* MULTI TOUCH ACTION:

C:\Users\User\Desktop\1.PNG

* HOW TI CLICK ON HOME BUTTON?

driver.sendKeyEvent(AndroidKeyCode.HOME);

* HOW TO HANDLE ALERT BOX?

USING TRY AND CATCH.

* HOW TO USE UI AUTOMATOR LOCATOR?

|  |
| --- |
|  |
|  | ---------- SYNTAX FOR USING UI AUTOMATOR ------------------------  driver.findElement(MobileBy.AndroidUIAutomator("")).click(); --> use this for UIAutomator Locator; |
|  |  |
|  |  |
|  | ---------- SYNTAX FOR UIAUTOMATOR ATTRIBUTE----------------------- |
|  | driver.findElement(MobileBy.AndroidUIAutomator("attribute(value)"); |
|  | ---------- USE BACKSLASH IF YOU HAVE "(DOUBLE QUOTATION) INSIDE "(DOUBLE QUOTATION); |
|  | driver.findElement(MobileBy.AndroidUIAutomator("attribute(\"value\")"); |
|  |  |
|  | --------- SYNATX FOR UIAUTOMATOR PROPERTIES ----------------------- |
|  | driver.findElements(MobileBy.AndroidUIAutomator("new UiSelector().clickable(true)")); --> clickable is a property; |
|  | \*/ |

* IMPORTANT ANDROID METHODS

|  |
| --- |
| /\* |
|  | \* Important Appium Methods; |
|  | \*/ |
|  |  |
|  | public class ImportantAppiumMethods extends BaseConfig { |
|  |  |
|  | public static void main(String[] args) throws MalformedURLException { |
|  |  |
|  | AndroidDriver<AndroidElement> driver = capabilities(); // calling the baseconfig driver; |
|  | // To see the current activity of the app; |
|  | System.out.println(driver.currentActivity()); |
|  |  |
|  | // To see which specific mode/app you are in? Native, Hybrid or WebView? |
|  | System.out.println(driver.getContext()); |
|  |  |
|  | // To see the rotating position of the current app; |
|  | System.out.println(driver.getOrientation()); |
|  |  |
|  | //driver.isLocked(); |
|  |  |
|  | // To hide the keyboard if it is open on the screen; |
|  | //driver.hideKeyboard(); |
|  |  |
|  | // To perform Back, Backspace, Home or keyboard specific methods; |
|  | driver.findElement(MobileBy.AndroidUIAutomator("text(\"Views\")")).click(); |
|  | driver.pressKeyCode(AndroidKeyCode.BACK); |
|  |  |
|  |  |
|  |  |
|  | } |
|  |  |
|  | } |

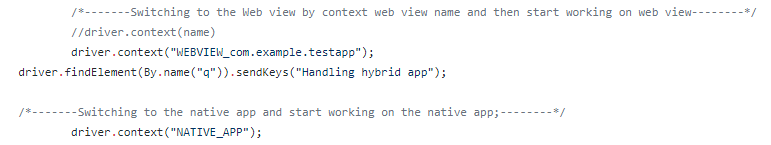
* HOW TO CHECK WHICH CONTEXT VIEW ARE YOU WORKING ON?

driver.getConText();

getContextHandles() – to see how many cotext are available in the app;

Set<String> s=driver.getContextHandles()

* HOW TO SWITCH FROM NATIVE TO WEB APP?



* IOS DRIVERS AND LOCATORS

|  |
| --- |
|  |
|  | IOSDriver<IOSElement>driver=ios\_base\_Config();  driver.findElementByAccessibilityId("Alert Views").click(); // Click on alert using powerful findElementByAccessibilityId; |
|  | driver.findElementByXPath("//\*[@name='Text Entry']").click(); // Click on text using xpath; |
|  | driver.findElementByClassName("XCUIElementTypeOther").sendKeys("hello"); //using class name; |
|  | driver.findElementByName("OK").click(); // using name locator; |

* HOW TO INSTALL AND REMOVE APP?
* *// Java INSTALL APP*
* Map<String, Object> params = **new** HashMap<>();
* params.put("app", "http://example.com/myapp.ipa");
* js.executeScript("mobile: installApp", params);
* MultiTouch/DOUBLE TAP

*MultiTouch* objects are collections of TouchActions.

MultiTouch gestures only have two methods, add, and perform.

add is used to add another TouchAction to this MultiTouch.

When perform is called, all the TouchActions which were added to the MultiTouch are sent to appium and performed as if they happened at the same time. Appium first performs the first event of all TouchActions together, then the second, etc.

Pseudocode example of tapping with two fingers:

action0 = **TouchAction**().**tap**(el)

action1 = **TouchAction**().**tap**(el)

**MultiAction**().**add**(action0).**add**(action1).**perform**()

How to go home button in ios app?

driver.runAppInBackground(Duration.ofSeconds(-1));

How to activate the app again in IOS?

driver.activateApp(bundle\_id);

How to switch one native app to another native app in ios?

static void launchSecurityXxxApp(AppiumDriver<AndroidElement> driver) throws MalformedURLException {

String appPackage="com.xxx.xxxx";

String appActivity="com.xxx.xxxx.Launchable";

Activity activity = new Activity(appPackage, appActivity);

activity.setStopApp(false);

((AndroidDriver<AndroidElement>) driver).startActivity(activity);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

}

public class SwitchAppTest {

    public AppiumDriver<MobileElement> driver = null;

    String calculatorAppPackageName = "com.sec.android.app.popupcalculator";

    String calculatorAppActivityName = "Calculator";

    String settingsAppPackageName = "com.android.settings";

    String settingsAppActivityName = "com.android.settings.GridSettings";

    @BeforeTest

    public void setupstart() throws MalformedURLException {

        DesiredCapabilities capabilities = DesiredCapabilities.android();

        capabilities.setCapability(MobileCapabilityType.AUTOMATION\_NAME, "Appium");

        capabilities.setCapability(MobileCapabilityType.PLATFORM\_NAME, "Android");

        capabilities.setCapability(MobileCapabilityType.DEVICE\_NAME, "4100b79b459381f7");

        capabilities.setCapability("appPackage", calculatorAppPackageName);

        capabilities.setCapability("appActivity", calculatorAppActivityName);

        driver = new AndroidDriver<MobileElement>(new URL("<http://localhost:4723/wd/hub>"), capabilities);

    }

    @Test

    public void calcTest1() throws Exception {

        //Multiply 2 numbers in calculator app

        driver.findElement(By.xpath("//android.widget.Button[@text='4']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@content-desc='Multiplication']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@text='4']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@content-desc='Equal']")).click();

        // launch settings App

        Activity activity = new Activity(settingsAppPackageName, settingsAppActivityName);

        activity.setStopApp(false);

        ((AndroidDriver<MobileElement>) driver).startActivity(activity);

        // Switch OFF WIFI

        driver.findElement(By.xpath("//android.widget.LinearLayout[@content-desc='Wi-Fi']")).click();

        driver.findElement(By.className("android.widget.Switch")).click();

        // Re launch calculator App

        activity = new Activity(calculatorAppPackageName, calculatorAppActivityName);

        activity.setStopApp(false);

        ((AndroidDriver<MobileElement>) driver).startActivity(activity);

        String result = driver.findElement(By.className("android.widget.EditText")).getText();

        System.out.println("Result : " + result);

    }

    @AfterTest

    public void tearDown() {

        driver.quit();

    }

}