**Mobile Gestures**

**Long Click Gesture**

This gesture performs long click action on the given element/coordinates. Available since Appium v1.19

We have to use this small piece of code in our Program

((JavascriptExecutor) driver).executeScript("mobile: longClickGesture", ImmutableMap.of(

"elementId", ((RemoteWebElement) element).getId()

));

**Scroll Gesture**

This gesture performs scroll gesture on the given element/area. Available since Appium v1.19

#### Returned value

The returned value is a boolean one and equals to true if the object can still scroll in the given direction

We have to use this piece of code

boolean canScrollMore = (Boolean) ((JavascriptExecutor) driver).executeScript("mobile: scrollGesture", ImmutableMap.of(

"left", 100, "top", 100, "width", 200, "height", 200,

"direction", "down",

"percent", 3.0

));

We can also use androidUiAutomator engine for scrolling

driver.findElement(AppiumBy.*androidUIAutomator*("new UiScrollable(new UiSelector()).scrollIntoView("+an+"(\""+av+"\"))"));

new UiScrollable(new UiSelector()).scrollIntoView(text(\"Visibility\"));

This piece of code help us to scroll a view within the mobile app till the element having some specific text

new UiScrollable(new UiSelector()) this will create an instance of UiScrollable class available in android automationtesting framework like UiAutomator 2 this class will help us to interact with scrollable views in a mobile app.

(new UiSelector()), Within the UiScrollable constructor, a new instance of UiSelector is created, UiSelector class help us to find element in in apps user interface

.scrollIntoView(text("Value")), it means we want to scroll within the view selected by UiSelector class, and it will scroll till the element for which text & value is mentioned

When we use this piece of code it will scroll through the scrollable view until it finds the element with text.

Ex-

**package** Project\_3\_Gestures;

**import** java.io.File;

**import** java.net.URL;

**import** java.time.Duration;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.testng.annotations.Test;

**import** com.google.common.collect.ImmutableMap;

**import** io.appium.java\_client.AppiumBy;

**import** io.appium.java\_client.android.AndroidDriver;

**import** io.appium.java\_client.remote.MobileCapabilityType;

**import** io.appium.java\_client.service.local.AppiumDriverLocalService;

**import** io.appium.java\_client.service.local.AppiumServiceBuilder;

**public** **class** Program\_2\_ScrollGesture {

@Test

**public** **void** locatorUsageTest() **throws** Throwable

{

//Starting server Programatically

File f=**new** File("C://Users// HI//AppData//Roaming//npm//node\_modules//appium//build//lib//main.js");

AppiumDriverLocalService sb=**new** AppiumServiceBuilder()

.withAppiumJS(f)

.withIPAddress("127.0.0.1").usingPort(4723).withTimeout(Duration.*ofSeconds*(300)).build();

sb.start();

DesiredCapabilities dc=**new** DesiredCapabilities();

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

dc.setCapability(MobileCapabilityType.***DEVICE\_NAME***, "Galaxy M30s");

dc.setCapability(MobileCapabilityType.***AUTOMATION\_NAME***, "UiAutomator2");

dc.setCapability(MobileCapabilityType.***UDID***, "RZ8M83ZJH2W");

dc.setCapability("appPackage", "io.appium.android.apis");

dc.setCapability("appActivity", ".ApiDemos");

URL u=**new** URL("http://localhost:4723");

AndroidDriver driver=**new** AndroidDriver(u,dc);

driver.manage().timeouts().~~implicitlyWait~~(20, TimeUnit.***SECONDS***);

driver.findElement(AppiumBy.*accessibilityId*("Views")).click();

//If we know where to scroll

driver.findElement(AppiumBy.*androidUIAutomator*("new UiScrollable(new UiSelector()).scrollIntoView(text(\"WebView\"));"));

//If we don't have prior idea

**boolean** canScrollMore;

**do**

{

canScrollMore = (Boolean) ((JavascriptExecutor) driver).executeScript("mobile: scrollGesture", ImmutableMap.*of*(

"left", 100, "top", 100, "width", 200, "height", 200,

"direction", "down",

"percent", 3.0

));

}

**while**(canScrollMore);

Thread.*sleep*(5000);

sb.stop();

}

}

**Swipe Gesture**

This gesture performs swipe gesture on the given element/area. Available since Appium v1.19

We have to use a piece of code in our program

((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.*of*(

"elementId",((RemoteWebElement) ele).getId(),

"direction", "left",

"percent", 0.75

));

Or

((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.of(

"left", 100, "top", 100, "width", 200, "height", 200,

"direction", "left",

"percent", 0.75

));

Ex-

**package** Project\_3\_Gestures;

**import** java.io.File;

**import** java.net.URL;

**import** java.time.Duration;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.openqa.selenium.remote.RemoteWebElement;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**import** com.google.common.collect.ImmutableMap;

**import** io.appium.java\_client.AppiumBy;

**import** io.appium.java\_client.android.AndroidDriver;

**import** io.appium.java\_client.remote.MobileCapabilityType;

**import** io.appium.java\_client.service.local.AppiumDriverLocalService;

**import** io.appium.java\_client.service.local.AppiumServiceBuilder;

**public** **class** Program\_3\_Swipe\_Gesture {

@Test

**public** **void** locatorUsageTest() **throws** Throwable

{

//Starting server Programatically

File f=**new** File("C://Users// HI//AppData//Roaming//npm//node\_modules//appium//build//lib//main.js");

AppiumDriverLocalService sb=**new** AppiumServiceBuilder()

.withAppiumJS(f)

.withIPAddress("127.0.0.1").usingPort(4723).withTimeout(Duration.*ofSeconds*(300)).build();

sb.start();

DesiredCapabilities dc=**new** DesiredCapabilities();

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

dc.setCapability(MobileCapabilityType.***DEVICE\_NAME***, "Galaxy M30s");

dc.setCapability(MobileCapabilityType.***AUTOMATION\_NAME***, "UiAutomator2");

dc.setCapability(MobileCapabilityType.***UDID***, "RZ8M83ZJH2W");

dc.setCapability("appPackage", "io.appium.android.apis");

dc.setCapability("appActivity", ".ApiDemos");

URL u=**new** URL("http://localhost:4723");

AndroidDriver driver=**new** AndroidDriver(u,dc);

driver.manage().timeouts().~~implicitlyWait~~(20, TimeUnit.***SECONDS***);

driver.findElement(AppiumBy.*accessibilityId*("Views")).click();

driver.findElement(AppiumBy.*accessibilityId*("Gallery")).click();

driver.findElement(AppiumBy.*accessibilityId*("1. Photos")).click();

WebElement ele = driver.findElement(AppiumBy.*xpath*("(//android.widget.ImageView)[1]"));

Assert.*assertEquals*(driver.findElement(AppiumBy.*xpath*("(//android.widget.ImageView)[1]")).getAttribute("focusable"), "true");

//Swipe

((JavascriptExecutor) driver).executeScript("mobile: swipeGesture", ImmutableMap.*of*(

"elementId",((RemoteWebElement) ele).getId(),

"direction", "left",

"percent", 0.75

));

Assert.*assertEquals*(driver.findElement(AppiumBy.*xpath*("(//android.widget.ImageView)[1]")).getAttribute("focusable"), "false");

}

}

**Drag & Drop Gesture**

This gesture performs drag action from the given element/coordinates to the given point. Available since Appium v1.19

We have to add a piece of code in our Program

((JavascriptExecutor) driver).executeScript("mobile: dragGesture", ImmutableMap.of(

"elementId", ((RemoteWebElement) element).getId(),

"endX", 100,

"endY", 100

));

Ex-

**package** Project\_3\_Gestures;

**import** java.io.File;

**import** java.net.URL;

**import** java.time.Duration;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.openqa.selenium.remote.RemoteWebElement;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**import** com.google.common.collect.ImmutableMap;

**import** io.appium.java\_client.AppiumBy;

**import** io.appium.java\_client.android.AndroidDriver;

**import** io.appium.java\_client.remote.MobileCapabilityType;

**import** io.appium.java\_client.service.local.AppiumDriverLocalService;

**import** io.appium.java\_client.service.local.AppiumServiceBuilder;

**public** **class** Program\_4\_Drag\_and\_Drop {

@Test

**public** **void** locatorUsageTest() **throws** Throwable

{

//Starting server Programatically

File f=**new** File("C://Users//SOUMYASANTA SAHOO//AppData//Roaming//npm//node\_modules//appium//build//lib//main.js");

AppiumDriverLocalService sb=**new** AppiumServiceBuilder()

.withAppiumJS(f)

.withIPAddress("127.0.0.1").usingPort(4723).withTimeout(Duration.*ofSeconds*(300)).build();

sb.start();

DesiredCapabilities dc=**new** DesiredCapabilities();

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

dc.setCapability(MobileCapabilityType.***DEVICE\_NAME***, "Galaxy M30s");

dc.setCapability(MobileCapabilityType.***AUTOMATION\_NAME***, "UiAutomator2");

dc.setCapability(MobileCapabilityType.***UDID***, "RZ8M83ZJH2W");

dc.setCapability("appPackage", "io.appium.android.apis");

dc.setCapability("appActivity", ".ApiDemos");

URL u=**new** URL("http://localhost:4723");

AndroidDriver driver=**new** AndroidDriver(u,dc);

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(30));

driver.findElement(AppiumBy.*accessibilityId*("Views")).click();

driver.findElement(AppiumBy.*accessibilityId*("Drag and Drop")).click();

WebElement ele = driver.findElement(AppiumBy.*id*("io.appium.android.apis:id/drag\_dot\_1"));

((JavascriptExecutor)driver).executeScript("mobile: dragGesture", ImmutableMap.*of*(

"elementId", ((RemoteWebElement) ele).getId(),

"endX", 626,

"endY", 585

));

Thread.*sleep*(2000);

String result = driver.findElement(AppiumBy.*id*("io.appium.android.apis:id/drag\_result\_text")).getText();

Assert.*assertEquals*(result, "Dropped!");

}

}

**Click Gesture**

This gesture performs click action on the given element/coordinates

We have to use a piece of code in our program

driver.executeScript("mobile: clickGesture", ImmutableMap.*of*(

"elementId", ((RemoteWebElement) ele).getId()

));

Or

driver.executeScript("mobile: clickGesture", ImmutableMap.*of*(

"x", x,

"y", y

));

**Pinch Open Gesture**

This gesture performs pinch-open gesture on the given element/area. Available since Appium v1.19

We have to use this piece of code in our program

((JavascriptExecutor) driver).executeScript("mobile: pinchOpenGesture", ImmutableMap.*of*(

"elementId", ((RemoteWebElement) ele).getId(),

"percent", 0.75

));

**Pinch Close Gesture**

This gesture performs pinch-close gesture on the given element/area. Available since Appium v1.19

We have to use this piece of code in our Program

((JavascriptExecutor) driver).executeScript("mobile: pinchCloseGesture", ImmutableMap.*of*(

"elementId", ((RemoteWebElement) ele).getId(),

"percent", 0.75

));