



Initialize

```
// Maven: io.appium.java_client
var desiredCapabilities = new DesiredCapabilities();
desiredCapabilities.setCapability("app", "Microsoft.WindowsCalculator_8wekyb3d8bbwe!App"); // for Universal Windows Platform apps
desiredCapabilities.setCapability("app", "C:\\Windows\\System32\\notepad.exe"); // for classic Windows apps
desiredCapabilities.setCapability("deviceName", "WindowsPC");
driver = new WindowsDriver<WindowsElement>(new URL("http://127.0.0.1:4723"), desiredCapabilities);
driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
```

Locators

```
driver.findElementById("42.333896.3.1");
driver.findElementByAccessibilityId("AppNameTitle");
driver.findElementByClassName("TextBlock");
driver.findElementByAccessibilityId("Views");
driver.findElementByName("Calculator");
driver.findElementByTagName("Text");
// Find multiple elements
driver.findElementsByClassName("TextBlock");
// Search for an element inside another element
driver.findElementByName("System")
    .findElementByName("Minimize");
```

Mouse & Keyboard Interactions

```
// Simple actions
element.click();
element.sendKeys("textToType");
element.clear();
// Chained actions
Actions actions = new Actions(_driver);
actions.moveToElement(element);
actions.moveByOffset(x, y);
actions.click();
actions.clickAndHold();
actions.release();
actions.contextClick();
actions.doubleClick();
actions.dragAndDrop(sourceElement, targetElement);
actions.dragAndDropBy(sourceElement, x, y);
actions.keyDown(Keys.SHIFT);
actions.keyUp(Keys.SHIFT);
actions.sendKeys("textToType");
actions.build().perform();
```

Touch & Pen Interactions

```
// Making a pen stroke
PointerInput device = new PointerInput(PointerInput.Kind.PEN, "pen");
Sequence sequence = new Sequence(device, 0);
sequence.addAction(device.createPointerMove(Duration.ZERO, PointerInput.Origin.fromElement(element), 0, 0));
sequence.addAction(device.createPointerDown(PointerInput.MouseButton.LEFT.asArg()));
sequence.addAction(device.createPointerMove(Duration.ZERO, PointerInput.Origin.fromElement(element), 10, 10));
sequence.addAction(device.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));
var actions = new ArrayList<Sequence>();
actions.add(sequence);
driver.perform(actions);
// Perform multitouch
PointerInput touch1 = new PointerInput(PointerInput.Kind.TOUCH, "finger");
Sequence touch1Sequence = new Sequence(touch1, 0);
touch1Sequence.addAction(touch1.createPointerMove(Duration.ZERO, PointerInput.Origin.fromElement(element), 50, -50));
touch1Sequence.addAction(touch1.createPointerDown(PointerInput.MouseButton.LEFT.asArg()));
touch1Sequence.addAction(touch1.createPointerMove(Duration.ofSeconds(1), PointerInput.Origin.fromElement(element), 80, -80));
touch1Sequence.addAction(touch1.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));
PointerInput touch2 = new PointerInput(PointerInput.Kind.TOUCH, "finger");
Sequence touch2Sequence = new Sequence(touch2, 0);
touch2Sequence.addAction(touch2.createPointerMove(Duration.ZERO, PointerInput.Origin.fromElement(element), -50, 50));
touch2Sequence.addAction(touch2.createPointerDown(PointerInput.MouseButton.LEFT.asArg()));
touch2Sequence.addAction(touch2.createPointerMove(Duration.ofSeconds(1), PointerInput.Origin.fromElement(element), -80, 80));
touch2Sequence.addAction(touch2.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));
var actions = new ArrayList<Sequence>();
actions.add(touch1Sequence);
actions.add(touch2Sequence);
driver.perform(actions);
```