**Method 1:**

import io.appium.java\_client.AppiumBy;  
import io.appium.java\_client.AppiumDriver;  
import io.appium.java\_client.android.AndroidDriver;  
import io.appium.java\_client.ios.IOSDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.interactions.PointerInput;  
import org.openqa.selenium.interactions.Sequence;  
import java.time.Duration;  
import java.util.Arrays;  
import static Resource.Swipe.SwipeDirection.\*;

public class Swipe {  
 private AppiumDriver driver;

public Swipe(AppiumDriver driver) {  
 this.driver = driver;  
 }

private WebElement ele;  
 private PointerInput finger = new PointerInput(PointerInput.Kind.TOUCH, "finger");  
 public void swipe(SwipeDirection direction) {

if (driver instanceof AndroidDriver) {  
 ele = driver.findElement(AppiumBy.xpath(TestElement.ANDROID\_GEN\_LOCATOR));  
 } else if (driver instanceof IOSDriver && (direction == SWIPE\_RIGHT || direction == SWIPE\_LEFT)) {  
 ele = driver.findElement(AppiumBy.xpath(TestElement.IOS\_GEN\_LOCATOR\_02));  
 } else {  
 ele = driver.findElement(AppiumBy.accessibilityId(TestElement.IOS\_GEN\_LOCATOR));  
 }

Sequence dragNDrop = new Sequence(finger, 1);  
 int startX, startY, endX, endY;  
 switch (direction) {  
 case SWIPE\_RIGHT:  
 startX = ele.getRect().x + (ele.getSize().width / 4);  
 startY = ele.getRect().y + (ele.getSize().height / 2);  
 endX = ele.getRect().x + (ele.getSize().width \* 3 / 4);  
 endY = ele.getRect().y + (ele.getSize().height / 2);  
 break;  
 case SWIPE\_LEFT:  
 startX = ele.getRect().x + (ele.getSize().width \* 3 / 4);  
 startY = ele.getRect().y + (ele.getSize().height / 2);  
 endX = ele.getRect().x + (ele.getSize().width / 4);  
 endY = ele.getRect().y + (ele.getSize().height / 2);  
 break;  
 case SWIPE\_DOWN:  
 startX = ele.getRect().x + (ele.getSize().width / 2);  
 startY = ele.getRect().y + (ele.getSize().height / 4);  
 endX = ele.getRect().x + (ele.getSize().width / 2);  
 endY = ele.getRect().y + (ele.getSize().height \* 3 / 4);  
 break;  
 case SWIPE\_UP:  
 startX = ele.getRect().x + (ele.getSize().width / 2);  
 startY = ele.getRect().y + (ele.getSize().height \* 3 / 4);  
 endX = ele.getRect().x + (ele.getSize().width / 2);  
 endY = ele.getRect().y + (ele.getSize().height / 4);  
 break;  
 default:  
 throw new IllegalArgumentException("Invalid swipe direction: " + direction);  
 }  
 dragNDrop.addAction(finger.createPointerMove(Duration.ofSeconds(0),  
 PointerInput.Origin.viewport(), startX, startY));  
 dragNDrop.addAction(finger.createPointerDown(PointerInput.MouseButton.LEFT.asArg()));  
 dragNDrop.addAction(finger.createPointerMove(Duration.ofMillis(700),  
 PointerInput.Origin.viewport(), endX, endY));  
 dragNDrop.addAction(finger.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));  
 driver.perform(Arrays.asList(dragNDrop));  
 }

public void tap(int x, int y) {  
 Sequence tap = new Sequence(finger, 1);  
 tap.addAction(finger.createPointerMove(Duration.ofMillis(0), PointerInput.Origin.viewport(), x, y));  
 tap.addAction(finger.createPointerDown(PointerInput.MouseButton.LEFT.asArg()));  
 tap.addAction(finger.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));  
 driver.perform(Arrays.asList(tap));  
 }  
 public enum SwipeDirection {  
 SWIPE\_RIGHT,  
 SWIPE\_LEFT,  
 SWIPE\_DOWN,  
 SWIPE\_UP  
 }  
}

**Method 2:**

public static void swipe(double startXFraction, double startYFraction, double endXFraction, double endYFraction) {

AndroidDriver driver = (AndroidDriver) DriverFactoryManager.getDriver();

Dimension size = driver.manage().window().getSize();

int startX = (int) (size.getWidth() \* startXFraction);

int startY = (int) (size.getHeight() \* startYFraction);

int endX = (int) (size.getWidth() \* endXFraction);

int endY = (int) (startY \* endYFraction);

PointerInput finger1 = new PointerInput(PointerInput.Kind.TOUCH, "finger1");

Sequence sequence = new Sequence(finger1, 1)

.addAction(finger1.createPointerMove(Duration.ZERO, PointerInput.Origin.viewport(), startX, startY))

.addAction(finger1.createPointerDown(PointerInput.MouseButton.LEFT.asArg()))

.addAction(new Pause(finger1, Duration.ofMillis(200)))

.addAction(finger1.createPointerMove(Duration.ofMillis(100), PointerInput.Origin.viewport(), endX, endY))

.addAction(finger1.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));

driver.perform(Collections.singletonList(sequence));

}

**Method 3:**

public static void swipe(double startXFraction, double startYFraction, double endXFraction, double endYFraction) {

AndroidDriver driver = (AndroidDriver) DriverFactoryManager.getDriver();

Dimension size = driver.manage().window().getSize();

int startX = (int) (size.getWidth() \* startXFraction);

int startY = (int) (size.getHeight() \* startYFraction);

int endX = (int) (size.getWidth() \* endXFraction);

int endY = (int) (startY \* endYFraction);

PointerInput finger1 = new PointerInput(PointerInput.Kind.TOUCH, "finger1");

Sequence sequence = new Sequence(finger1, 1)

.addAction(finger1.createPointerMove(Duration.ZERO, PointerInput.Origin.viewport(), startX, startY))

.addAction(finger1.createPointerDown(PointerInput.MouseButton.LEFT.asArg()))

.addAction(new Pause(finger1, Duration.ofMillis(200)))

.addAction(finger1.createPointerMove(Duration.ofMillis(100), PointerInput.Origin.viewport(), endX, endY))

.addAction(finger1.createPointerUp(PointerInput.MouseButton.LEFT.asArg()));

driver.perform(Collections.singletonList(sequence));

}