

# APK Installation and App Execution on Real & Virtual Devices

## 1. Objective

The objective of this activity is to demonstrate the successful installation of APK files on both a **real Android device** and a **virtual device (Android Emulator)**, and to validate basic application functionality by performing actions on **Dialer** and **Calculator** applications.

---

## 2. Environment Details

### 2.1 Software Setup

- **Operating System:** Windows
- **Java (JDK):** Installed and configured
- **Android Studio:** Installed
- **Android SDK & Platform Tools:** Configured
- **ADB (Android Debug Bridge):** Verified
- **Appium (Optional):** Installed for automation support

### 2.2 Hardware Setup

- **Real Android Device:** Connected via USB
  - **Virtual Device:** Android Emulator (AVD) created using Android Studio
- 

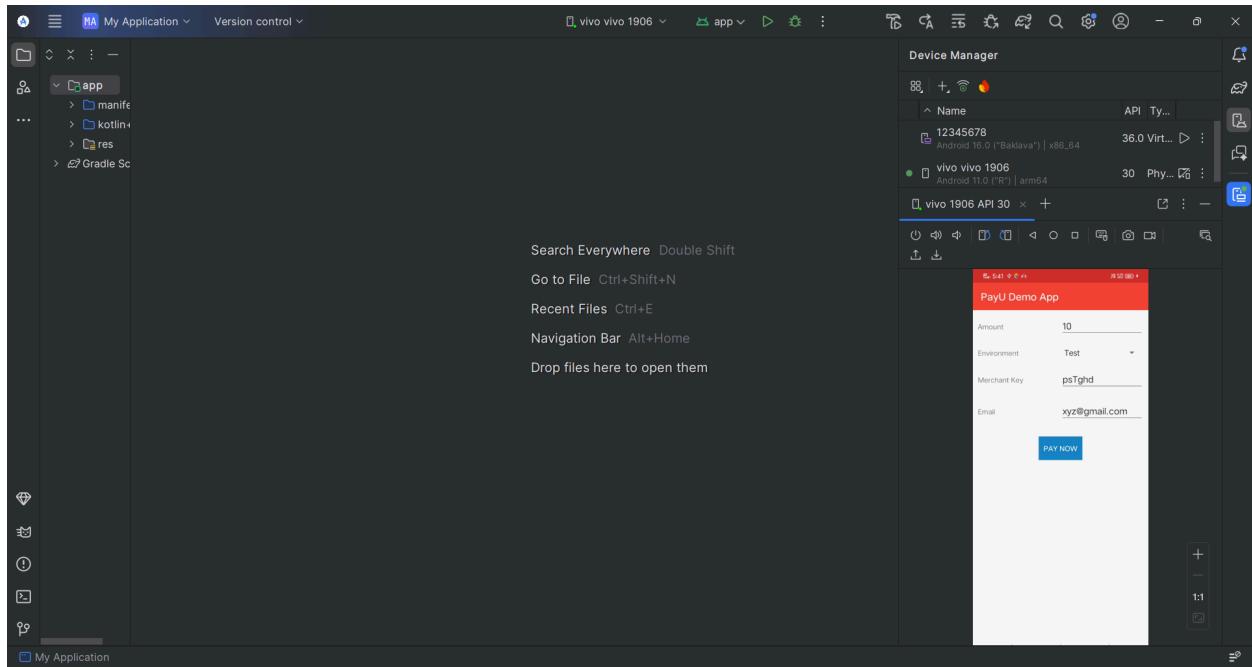
## 3. Device Configuration

### 3.1 Real Device Setup

- Enabled **Developer Options**
- Enabled **USB Debugging**
- Device successfully detected using ADB command:  
`adb devices`

## 3.2 Virtual Device (Emulator) Setup

- Android Virtual Device (AVD) created and launched
- Emulator verified as running and visible in ADB devices list



## 4. APK Installation

### 4.1 Installing APK on Real Device

Steps followed:

1. Connected the real device via USB
2. Verified device connection using ADB
3. Installed APK using the command:  
`adb install <apk-file-path>`
4. Installation completed successfully without errors

**Result:** APK installed successfully on the real device

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The "Package Explorer" view shows a project named "Newexample" with packages "src/main/java" and "src/test/java". Under "src/test/java/Demo", there are three files: "Apkinstallation.java", "CalculatorTest.java", and "Caldemo.java".
- Code Editor:** The "Apkinstallation.java" file is open, displaying Java code for automating an Android application using the UiAutomator2 API. The code includes imports for `java.net.URI` and `UiAutomator2Options`, sets up options for the emulator (platform name: "Android", device name: "vินчестер 1906"), and installs an APK from the local file system ("C:\\Users\\pavit\\apk files\\SampleApp.apk"). It then creates a `AndroidDriver` instance and prints "Application Started" to the console.
- Console Output:** The "Console" view shows the output of the executed Java application. It includes the command-line arguments, the Java runtime environment information, and the message "Application Started".

## 4.2 Installing APK on Virtual Device (Emulator)

Steps followed:

1. Started the Android Emulator
2. Verified emulator connection using ADB
3. Installed APK using the command:  
adb install <apk-file-path>
4. APK installation completed successfully

**Result:** APK installed successfully on the virtual device

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "Newexample". The "src/test/java/Demo" package contains three files: Apkinstallation.java, CalculatorTest.java, and Caldemo.java.
- Code Editor:** The file "Apkinstallation.java" is open, showing Java code for automating an Android application. The code imports `java.net.URI` and defines a class `Apkinstallation` with a `main` method. It uses `UiAutomator2Options` to set platform name to "Android", device id to "12345678", automation name to "UiAutomator2", platform version to "10", and app path to "C:\\Users\\pavit\\apk files\\SampleApp.apk". It then creates a URL, sets up an `AndroidDriver`, sleeps for 5000ms, and prints "Application Started" if the session ID is not null.
- Console:** Shows the output of the application execution. It includes the command `Apkinstallation [Java Application] C:\Users\pavit\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86\_64\_21.0.9.v20251105-0741\jre\bin\javaw.exe (03-Feb-2026, 5:43:16 pm - 5:43:35 pm) Application Started` and the message "Application Started".

## 5. Application Execution & Validation

### 5.1 Dialer Application Execution

- Opened the Dialer application on the emulator
- Entered phone numbers using the dial pad
- Verified UI responsiveness and functionality

**Result:** Dialer application worked successfully on the virtual device

#### 5.1.2 Dialer on Real Device

- Opened the Dialer application on the real device
- Entered phone numbers using the dial pad
- Verified proper number input and screen response

**Result:** Dialer application worked successfully on the real device

```

eclipse-workspace - Newexample/src/test/java/Demo/Caldemo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X Apkinstallation.java Caldemo.java X
1 package Demo;
2 import java.net.MalformedURLException;
3
4 public class Caldemo {
5     public static void main(String[] args) throws MalformedURLException, InterruptedException {
6         DesiredCapabilities dc = new DesiredCapabilities();
7         dc.setCapability("platformName", "Android");
8         dc.setCapability("appium:automationName", "uiautomator2");
9         dc.setCapability("appium:deviceName", "Xiaomi 2106111981");
10        dc.setCapability("appium:deviceName", "vivo 1906");
11        dc.setCapability("appium:appPackage", "com.android.dialer");
12        dc.setCapability("appium:newCommandTimeout", 300);
13        URL url = URL.create("http://127.0.0.1:4723/").toURL();
14        AndroidDriver driver = new AndroidDriver(url, dc);
15        Thread.sleep(2000);
16        // driver.findElement(By.id("com.google.android.dialer:id/tab_call_history")).click();
17        // driver.findElement(By.id("com.google.android.dialer:id/dialpad_fab")).click();
18        // for (Integer i = 0; i < 10; i++) {
19        //     driver.findElement(By.id("com.google.android.dialer:id/one")).click();
20        //     Thread.sleep(500);
21        // }
22        driver.findElement(By.id("com.google.android.dialer:id/dialpad_voice_call_button")).click();
23        Thread.sleep(4000);
24        driver.findElement(By.xpath("//android.view.View[@content-desc='End call']/..")).click();
25        driver.findElement(By.id("com.google.android.dialer:id/incall_end_call")).click();
26        driver.quit();
27        System.out.println("App completed");
28    }
29}
30
31
32
33
34
35
36
37
38
39

```

Console X Eclipse IDE for Java Developers 2026-03 M2  
<terminated> Caldemo [Java Application] C:\Users\pavit\p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86\_64\_21.0.9.v20251105-0741\jre\bin\javaw.exe (03-Feb-2026, 5:46:40 pm – 5:46:55 pm elas  
App completed

## 5.2 Calculator Application Execution (Real Device)

- Opened the Calculator application on the real device
- Performed basic arithmetic operations:
  - Addition
  - Subtraction
  - Multiplication
  - Division
- Verified correct results and smooth operation

**Result:** Calculator application worked successfully on the real device