

Automation of Android Calculator and Phone Dialer using Appium

1. Objective

The objective of this activity is to automate two default Android system applications on a real device using Appium:

- Calculator Application — to perform and verify an arithmetic operation
- Phone Dialer Application — to dial a phone number and initiate a call

The automation is implemented using **Java**, **Appium** and executed from **Eclipse IDE**.

2. Environment Setup

Software Used

- Windows Operating System
- Java JDK
- Android Studio (SDK & Platform Tools)
- Appium Server
- Appium Inspector
- Eclipse IDE (Maven Project)

Hardware Used

- Real Android device connected via USB
- USB Debugging enabled

- Device verified using `adb devices`
-

3. Common Desired Capabilities Used

Both programs use the following common configuration:

- Platform: Android
 - Automation Engine: UiAutomator2
 - Device identified using UDID
 - Appium server running at `http://127.0.0.1:4723`
 - New command timeout set to 60 seconds
-

Part A — Automating Android Calculator

4. Application Details Identified

Using Appium Inspector, the following details were captured for the calculator:

- **App Package:** `com.coloros.calculator`
- **App Activity:** `com.android.calculator2.Calculator`

The resource IDs of the calculator buttons were identified:

- Digit 8 button
 - Plus (+) operator button
 - Digit 2 button
 - Equal (=) button
 - Result display field
-

5. Automation Steps Performed

The program performs the following sequence:

1. Launches the Calculator application using its package and activity name
 2. Waits for the application to load
 3. Clicks on digit **8**
 4. Clicks on the **Plus (+)** operator
 5. Clicks on digit **2**
 6. Clicks on the **Equal (=)** button
 7. Reads the result displayed on the calculator screen
 8. Compares the result with the expected value **10**
 9. Prints **Pass** if correct, otherwise **Fail**
 10. Closes the Appium session
-

6. Result of Calculator Automation

The calculator successfully performs the operation **8 + 2**.

The program validates the result by reading the displayed value and confirms the correctness of the operation.

Part B — Automating Android Phone Dialer

7. Application Details Identified

Using Appium Inspector, the following dialer details were obtained:

- **App Package:** `com.google.android.dialer`
- **App Activity:**
`com.google.android.dialer.extensions.GoogleDialtactsActivity`

The resource IDs of dial pad buttons and call button were identified.

8. Automation Steps Performed

The program performs the following sequence:

1. Launches the Phone Dialer application
2. Waits for the application to load
3. Opens the dial pad
4. Clicks the number buttons to enter the phone number:
 - 0 8 7 8 9 1 6 8 4 7 3
5. Clicks on the call button to initiate the call
6. Waits for a few seconds
7. Closes the Appium session

9. Result of Dialer Automation

The dialer application successfully:

- Opens the dial pad
- Enters the complete phone number automatically
- Initiates the call

This confirms successful interaction with the dialer application through automation.

10. Conclusion

This activity demonstrates:

- Launching Android system applications using Appium
- Identifying UI elements using Appium Inspector
- Using resource-id locators for accurate element identification
- Performing real user actions such as calculations and dialing numbers

- Validating application behavior through automation on a real device

Both the **Calculator** and **Phone Dialer** applications were automated successfully using Java and UiAutomator2.