Lab: Appium Automated Tests

1. "Summator" Android App

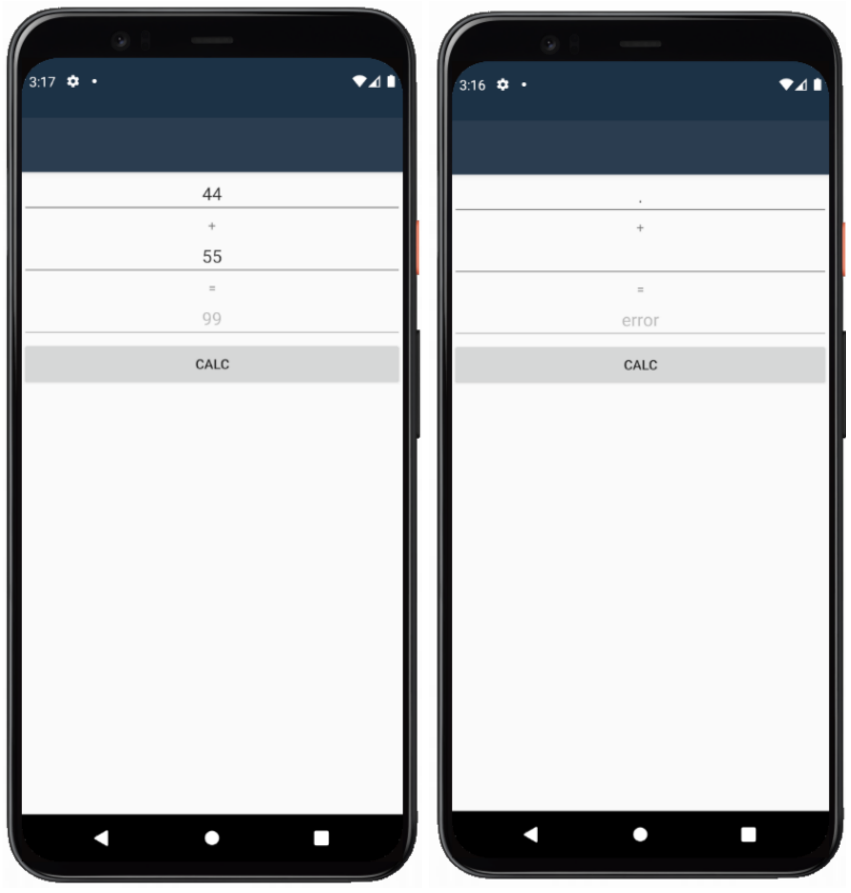
You are given the following sample **Android mobile app** "Summator": https://github.com/nakov/AndroidApp-Summator.

You can get its .apk file from here: https://github.com/nakov/AndroidApp-Summator/releases/tag/v1.0.

You also have the .apk file added to the lecture's resources.

2. The Automated Testing Scenario

1. Open the **`Summator` app.**
2. Test with **valid** and **invalid data.**
   - With valid data: assert that result is **correct.**
   - With invalid data: assert that "**`error`**" is displayed in the result field

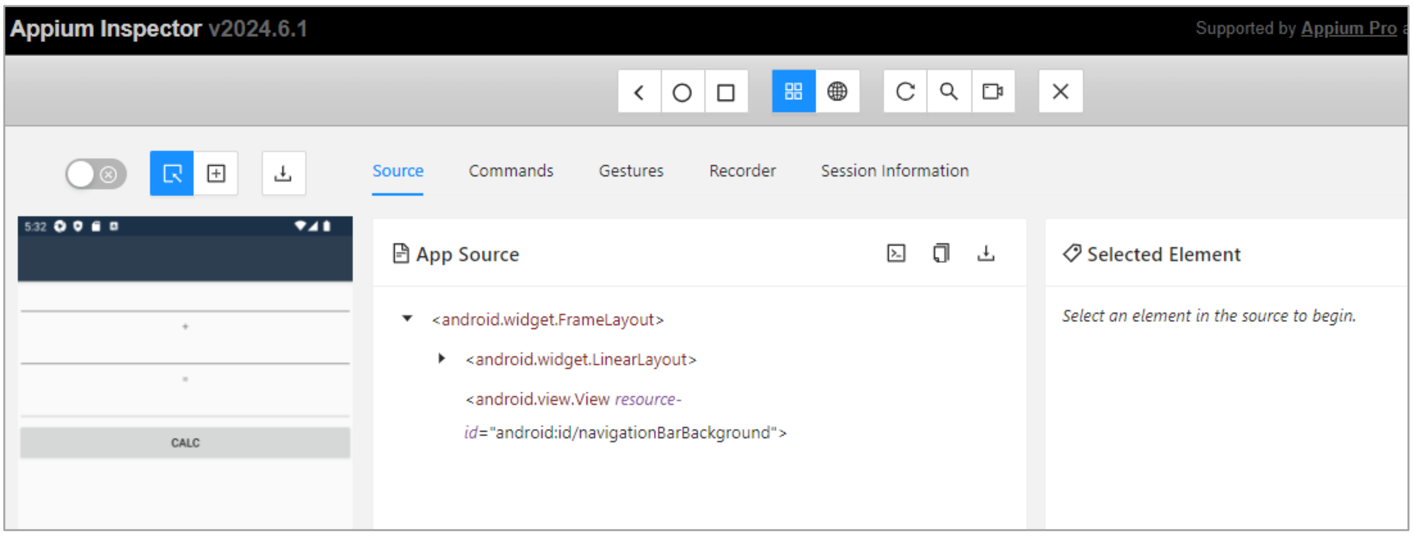


3. Configure Appium Inspector

- **Start** Appium Server (if you're using the Web version of Appium Inspector use `--allow-cors`)

```
appium      OR    appium --allow-cors
```

- Start your AVD in a separate CMD window
- Make sure that the Application for testing is installed on your AVD (drag & drop to install)
- Provide the **host and port of Appium server** in Appium Inspector (**`http://127.0.0.1:4723`**)
- Add the Desired Capabilities of the mobile device or emulator connected to the system
  - **`automationName`** – in our case **`UiAutomator2`**
  - **`platformName`** - get it using command **`appium driver list`**
  - **`platformVersion`** - get it using command **`adb shell getprop ro.build.version.release`**
  - **`appium:deviceName`** – get it using command **`adb devices`**
  - **`appium:app`** – path to the.apk file for testing on your computer



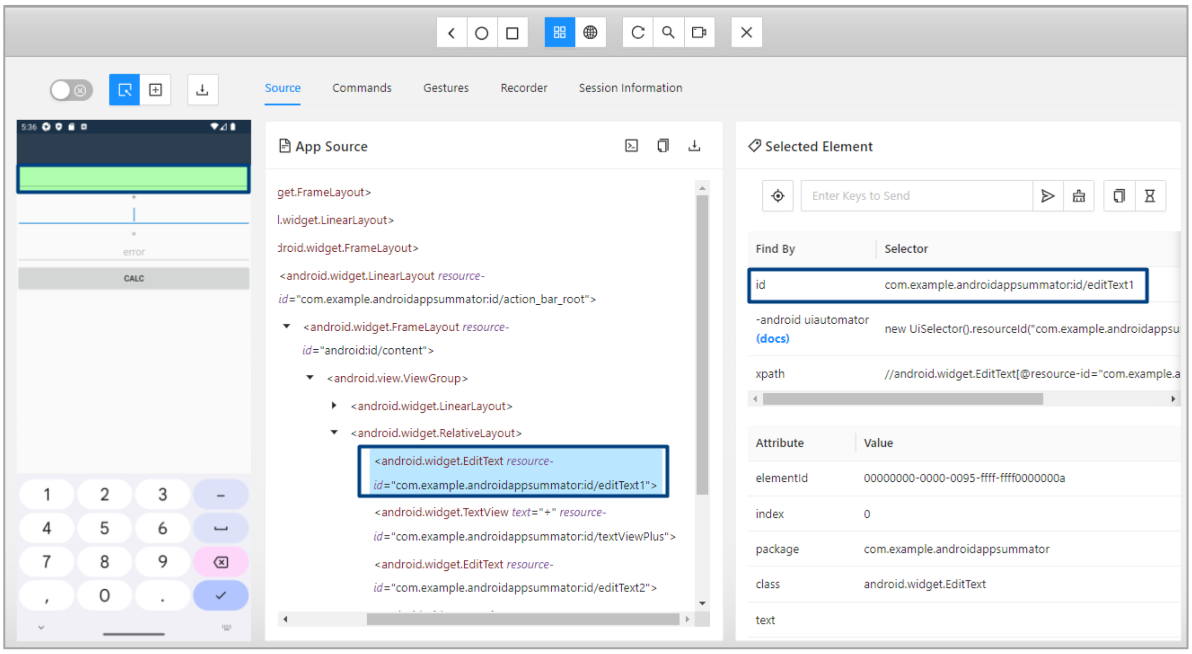
```
▾ <android.widget.FrameLayout>
  ▸ <android.widget.LinearLayout>
    <android.view.View resource-
    id="android:id/navigationBarBackground">
```

Select an element in the source to begin.

Whatever happens on Appium Inspector happens on the AVD as well.

4. Get the ids of the elements

You can get the id of any element you need. Click on the element e.g., on the first number field, and its id will be shown under **"Selected Element"**:



```
get.FrameLayout>
l.widget.LinearLayout>
droid.widget.FrameLayout>
 <android.widget.LinearLayout resource-
 id="com.example.androidappsummator:id/action_bar_root">
  ▾ <android.widget.FrameLayout resource-
    id="android:id/content">
    ▾ <android.view.ViewGroup>
      ▸ <android.widget.LinearLayout>
      ▾ <android.widget.RelativeLayout>
          <android.widget.EditText resource-
          id="com.example.androidappsummator:id/editText1">
          <android.widget.TextView text="+" resource-
          id="com.example.androidappsummator:id/textViewPlus">
          <android.widget.EditText resource-
          id="com.example.androidappsummator:id/editText2">
```

Find By	Selector
id	com.example.androidappsummator:id/editText1
-android uiautomator (docs)	new UiSelector().resourceId("com.example.androidappsu
xpath	//android.widget.EditText[@resource-id="com.example.a

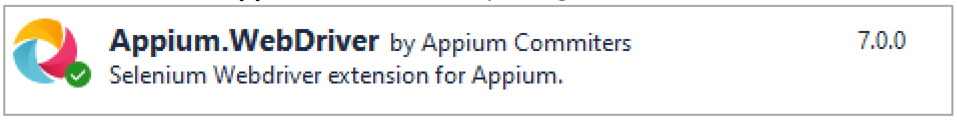
Attribute	Value
elementId	00000000-0000-0095-ffff-ffff0000000a
index	0
package	com.example.androidappsummator
class	android.widget.EditText
text	

You should do the same to get the IDs of all the elements you need.

5. Write Tests in VS

Let's write some tests.

- Open **`Visual Studio`** and create a new **`C# NUnit Test Project`**.
- Install the **`Appium.WebDriver`** package from NuGet.



Appium.WebDriver by Appium Commiters 7.0.0
Selenium Webdriver extension for Appium.

- Import the necessary namespaces.

```
using OpenQA.Selenium.Appium;
using OpenQA.Selenium.Appium.Android;
using OpenQA.Selenium.Appium.Service;
```

- Create a SummatorTests class and mark it with [TestFixture] to indicate that it contains tests.
- Declare the private fields for the Android driver and Appium local service.

```
private AndroidDriver driver;
private AppiumLocalService appiumLocalService;
```

- Create a setup method to initialize the Appium server and Android driver.

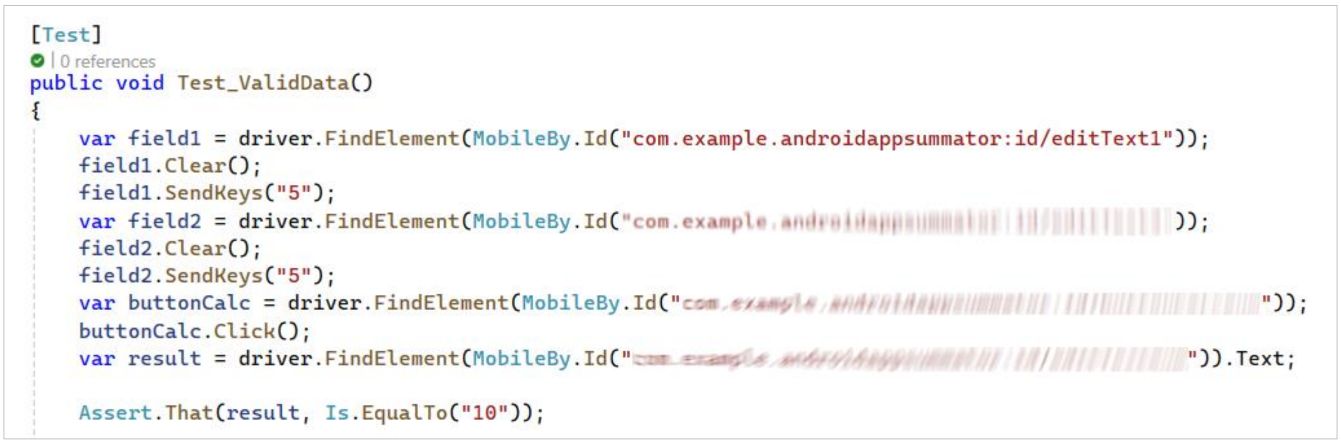
```
[OneTimeSetUp]
0 references
public void RunBeforeAnyTests()
{
    // Starts Appium server no need for you to start it via CMD
    appiumLocalService = new AppiumServiceBuilder()
        .WithIPAddress("127.0.0.1")
        .UsingPort(4723)
        .Build();
    appiumLocalService.Start();

    var androidOptions = new AppiumOptions();
    androidOptions.PlatformName = "Android";
    androidOptions.AutomationName = "UIAutomator2";
    androidOptions.DeviceName = "Pixel7";
    androidOptions.App = @"D:\com.example.androidappsummator.apk";

    driver = new AndroidDriver(appiumLocalService.ServiceUrl, androidOptions);
}
```

  - **[OneTimeSetUp]:** This attribute indicates that the RunBeforeAnyTests method should be run once before any of the tests in the class.
  - **Appium Service Setup:** Creates and starts an Appium server locally on IP 127.0.0.1 and port 4723.
  - **Android Options:** Configures the options for the Android driver:
  - **PlatformName:** Specifies the platform as Android.
  - **AutomationName:** Specifies the automation engine (UIAutomator2).
  - **DeviceName:** Names the device/emulator.
  - **App:** Path to the APK file of the application to be tested.
  - **Driver Initialization:** Instantiates the AndroidDriver with the service URL and the options defined.

- Create a tear down method to clean up after the tests.

```
[OneTimeTearDown]
0 references
public void RunAfterAnyTests()
{
    driver?.Quit();
    driver?.Dispose();
    appiumLocalService.Dispose();
}
```

- Write a test method to validate functionality **with valid data**.

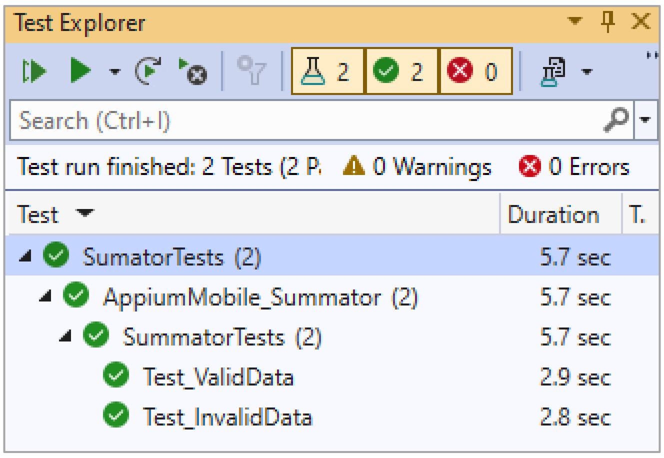


```
[Test]
0 references
public void Test_ValidData()
{
    var field1 = driver.FindElement(MobileBy.Id("com.example.androidappsummator:id/editText1"));
    field1.Clear();
    field1.SendKeys("5");
    var field2 = driver.FindElement(MobileBy.Id("com.example.[redacted]));
    field2.Clear();
    field2.SendKeys("5");
    var buttonCalc = driver.FindElement(MobileBy.Id("[redacted]"));
    buttonCalc.Click();
    var result = driver.FindElement(MobileBy.Id("[redacted]")).Text;

    Assert.That(result, Is.EqualTo("10"));
```

  - Use **`MobileBy.Id`** to get the **first number field** by its **`id`**, which you got from **`Appium Inspector`**.
  - **Clear** the field, otherwise if it is not empty, new text will be just added after the old one.
  - Next, **write** "**`5`**" in the field.
  - Do the same with the **second number field**, where you should type "**`5`**".
  - Press the **`[Calculate]`** button and assert that the text in the **result field** is "**`10`**".

- Write the test with **invalid data** and "**`error`**" as a result, as well.
  - You may type "**`.`**" in the first number field and leave the second field **empty**.
  - Then, "**`error`**" will be displayed.

6. Run the Tests



```
Test Explorer
2   2   0
Search (Ctrl+I)
Test run finished: 2 Tests (2 P.   0 Warnings   0 Errors
Test                                Duration   T.
◢ SumatorTests (2)                  5.7 sec
  ◢ AppiumMobile_Summator (2)       5.7 sec
    ◢ SummatorTests (2)             5.7 sec
        Test_ValidData              2.9 sec
        Test_InvalidData            2.8 sec
```