Group_1:
Hrushikesh Pawar
Anupam Pandey
HarshWardhan Sharma
Manish Singh
Syed Mohammed Tahir
Krishna Zanwar
Asha Julupalli

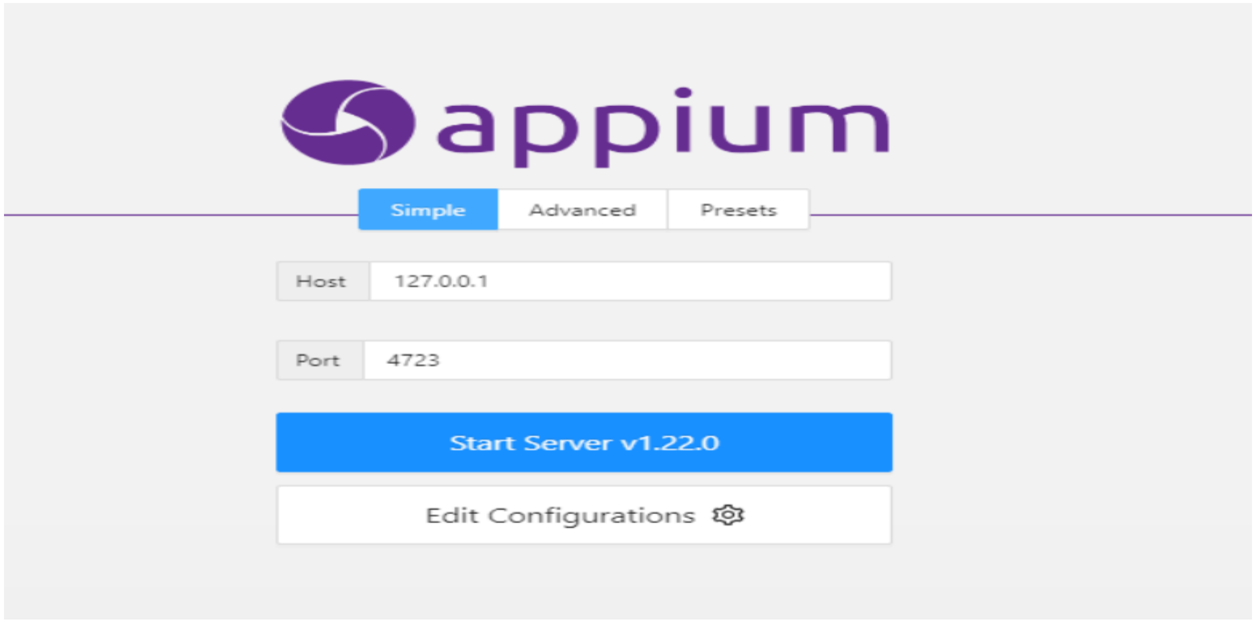
Comprehensive Guide to Setting Up and Running Appium with Capabilities and App Installation

This guide will walk you through the entire process of setting up Appium, configuring capabilities, starting the server, using the Appium Inspector, and running a test session to install an app on an Android emulator.

1. Opening the Appium Server

1. Launch the Appium Server application:

   Open the Appium Desktop app by clicking on its icon or running it from the start menu



Host 127.0.0.1

Port 4723

2. Set Remote Host and Port:

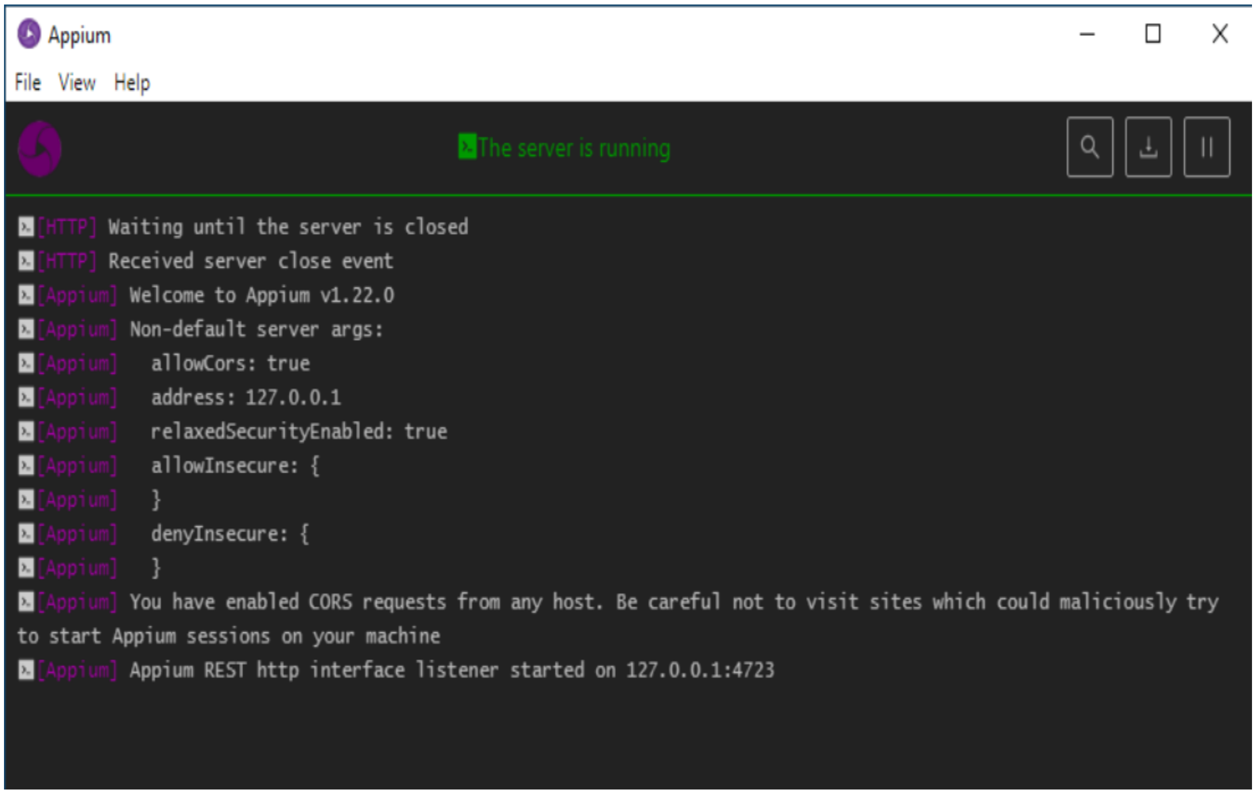
Remote Host: This should be “127.0.0.1”, which indicates that the server is running on your local machine.

Remote Port: The default port for Appium server is “4723”. Ensure this port is not being used by any other application.

Remote Path: This should be “/wd/hub”. This is the default path used by Appium to handle WebDriver requests.

3. Start the Server:

Click on the 'Start Server' button to start the Appium server. You should see logs appearing in the Appium Desktop window, indicating that the server is running. If there are any errors, they will also appear here.



```
[HTTP] Waiting until the server is closed
[HTTP] Received server close event
[Appium] Welcome to Appium v1.22.0
[Appium] Non-default server args:
[Appium]   allowCors: true
[Appium]   address: 127.0.0.1
[Appium]   relaxedSecurityEnabled: true
[Appium]   allowInsecure: {
[Appium]   }
[Appium]   denyInsecure: {
[Appium]   }
[Appium] You have enabled CORS requests from any host. Be careful not to visit sites which could maliciously try
to start Appium sessions on your machine
[Appium] Appium REST http interface listener started on 127.0.0.1:4723
```

2. Open Appium Inspector

1. Launch Appium Inspector:

   Open the Appium Inspector tool from the Appium Desktop app. This tool allows you to inspect the elements of your app and interact with it, which is useful for writing test scripts.

2. Connect to the Appium Server:

   Enter the server address: “127.0.0.1”.

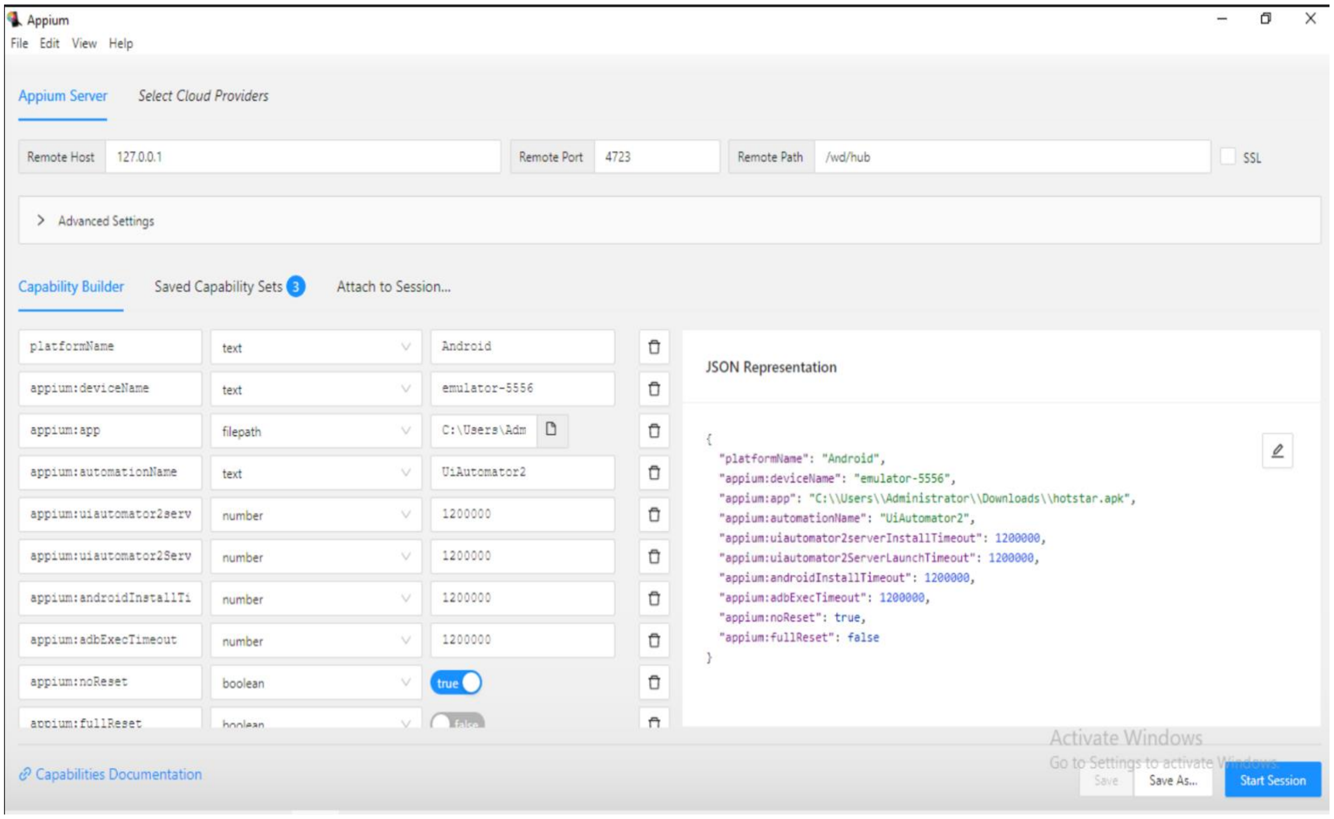
   Enter the server port: “4723”.

   Ensure the path is “/wd/hub”.

3. Setup Capabilities in Appium Inspector

1. Open Capability Builder:

   In the Appium Inspector tool, go to the “Capability Builder” section. Here you will define the capabilities required to run your test.



Remote Host	127.0.0.1
Remote Port	4723
Remote Path	/wd/hub
SSL	

platformName	text	Android
appium:deviceName	text	emulator-5556
appium:app	filepath	C:\Users\Adm
appium:automationName	text	UiAutomator2
appium:uiautomator2serv	number	1200000
appium:uiautomator2Serv	number	1200000
appium:androidInstallTi	number	1200000
appium:adbExecTimeout	number	1200000
appium:noReset	boolean	true
appium:fullReset	boolean	false

JSON Representation

```
{
  "platformName": "Android",
  "appium:deviceName": "emulator-5556",
  "appium:app": "C:\\Users\\Administrator\\Downloads\\hotstar.apk",
  "appium:automationName": "UiAutomator2",
  "appium:uiautomator2serverInstallTimeout": 1200000,
  "appium:uiautomator2ServerLaunchTimeout": 1200000,
  "appium:androidInstallTimeout": 1200000,
  "appium:adbExecTimeout": 1200000,
  "appium:noReset": true,
  "appium:fullReset": false
}
```

Activate Windows
Go to Settings to activate Windows.

2. Define Each Capability:

platformName:

Type: “text”

Value: “"Android"“

Explanation: Specifies the mobile platform name. In this case, it's Android. This tells Appium to prepare the environment for Android automation.

appium:deviceName:

Type: “text”

Value: “"emulator5556"“

Explanation: Specifies the name of the device or emulator. Here, “emulator5556” is the identifier of the running Android emulator. You can find this by running “adb devices” in the command line.

appium:app:

Type: “filepath”

Value: “"C:\\Users\\Administrator\\Downloads\\hotstar.apk"“

Explanation: Path to the APK file that needs to be tested. This is the app that Appium will install on the emulator. Ensure the path is correct and the file exists.

appium:automationName:

Type: “text”

Value: “"UiAutomator2"“

Explanation: Specifies the automation engine to use. “UiAutomator2” is recommended for Android and provides better support for newer Android versions.

appium:uiautomator2ServerInstallTimeout:

Type: “number”

Value: “1200000”

Explanation: Timeout (in milliseconds) for installing the UiAutomator2 server. Setting a high value ensures there's enough time for installation, especially for larger APKs.

appium:uiautomator2ServerLaunchTimeout:

Type: “number”

Value: “1200000”

Explanation: Timeout (in milliseconds) for launching the UiAutomator2 server. This allows enough time for the server to start and be ready for commands.

appium:androidInstallTimeout:

Type: “number”

Value: “1200000”

Explanation: Timeout (in milliseconds) for installing the Android app. Increasing this value helps to avoid timeouts during the app installation process.

appium:adbExecTimeout:

Type: “number”

Value: “1200000”

Explanation: Timeout (in milliseconds) for ADB (Android Debug Bridge) commands. Increasing this value can help avoid timeouts during lengthy operations, such as large file transfers or complex command executions.

appium:noReset:

Type: “boolean”

Value: “true”

Explanation: Prevents the app from being reset between sessions. Useful if you want to maintain the app's state between tests.

appium:fullReset:

Type: “boolean”

Value: “false”

Explanation: Ensures the app data is not cleared and the app is not uninstalled after the test. Useful for keeping the app in the same state between tests.

4. JSON Representation of Capabilities

The JSON representation of the capabilities is generated automatically on the right side of the screen as you enter values in the “Capability Builder”. This JSON is used to start the Appium session and should be copied if you need to run the session programmatically.

“““json

```
{
  "platformName": "Android",
  "appium:deviceName": "emulator5556",
  "appium:app": "C:\\Users\\Administrator\\Downloads\\hotstar.apk",
  "appium:automationName": "UiAutomator2",
  "appium:uiautomator2ServerInstallTimeout": 1200000,
  "appium:uiautomator2ServerLaunchTimeout": 1200000,
  "appium:androidInstallTimeout": 1200000,
  "appium:adbExecTimeout": 1200000,
  "appium:noReset": true,
  "appium:fullReset": false
}
```

“““

5. Starting the Appium Session

1. Start Session:

   Once all capabilities are set, click on “Start Session” in the Appium Inspector tool.

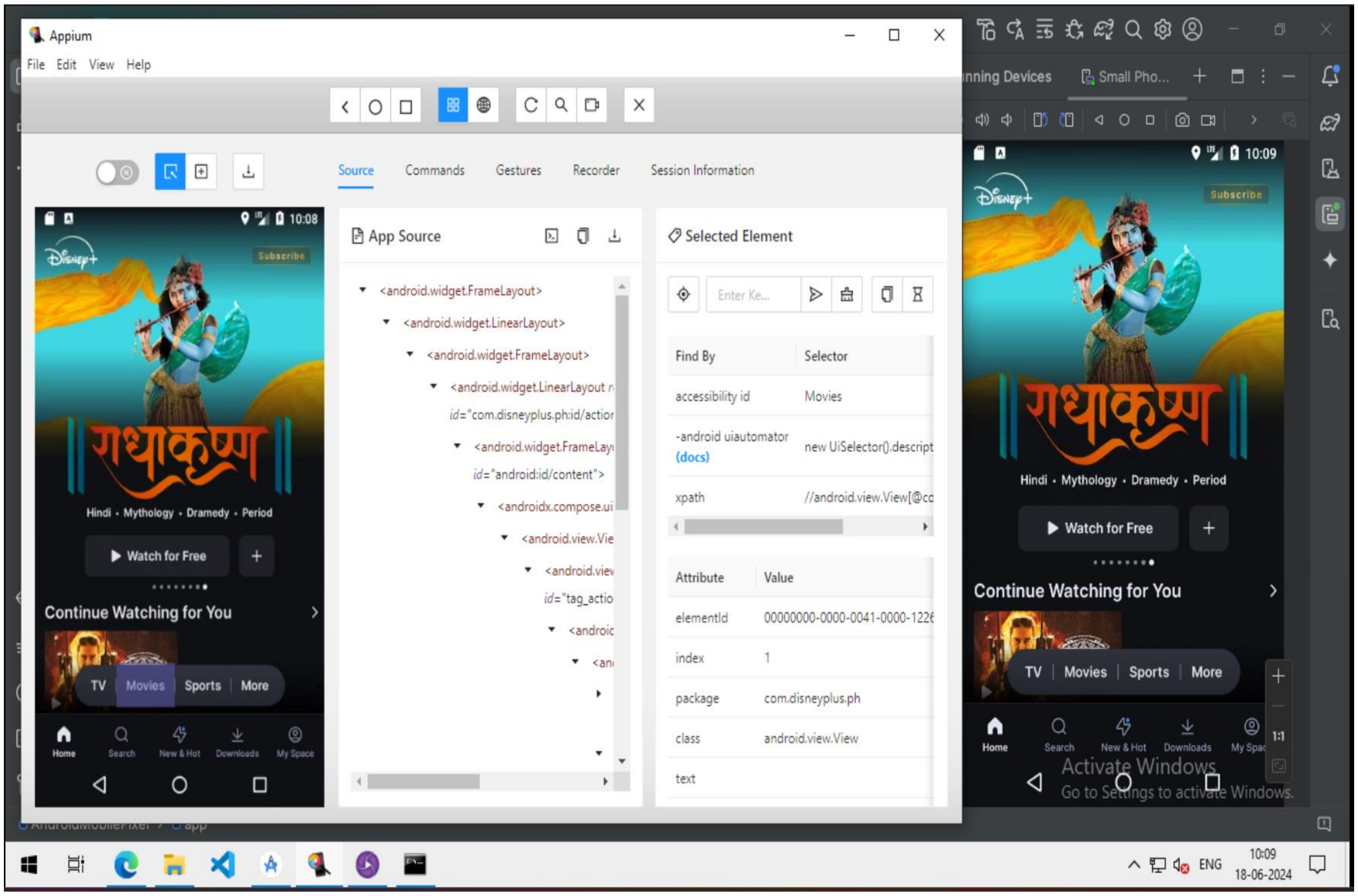
   Appium will then try to install the specified APK on the specified device/emulator and launch the app. Ensure the emulator is already running.

6. App Installation

1. Verify Installation:

   Appium will install the APK file located at the specified path (“C:\\Users\\Administrator\\Downloads\\hotstar.apk”) on the emulator with the device name “emulator5556”.

   The specified app should start on the emulator automatically.



```
<android.widget.FrameLayout>
  <android.widget.LinearLayout>
    <android.widget.FrameLayout>
      <android.widget.LinearLayout r
      id="com.disneyplus.ph:id/actior
        <android.widget.FrameLay
        id="android:id/content">
          <androidx.compose.ui
            <android.view.Vie
              <android.view
              id="tag_actio
                <androic
                  <an
```

Find By	Selector
accessibility id	Movies
-android uiautomator (docs)	new UiSelector().descript
xpath	//android.view.View[@cc

Attribute	Value
elementId	00000000-0000-0041-0000-122€
index	1
package	com.disneyplus.ph
class	android.view.View
text	

5. Identifying Elements using Appium Inspector

Inspect Elements:

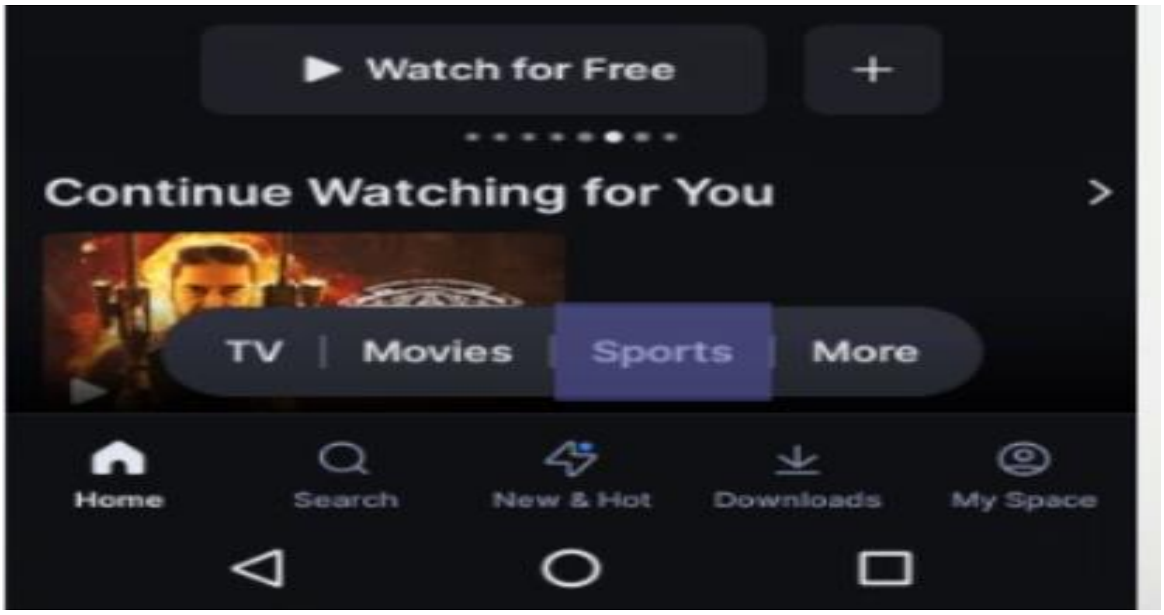
Once the app is launched on the emulator, the Appium Inspector tool will display the screen of the app along with its elements.

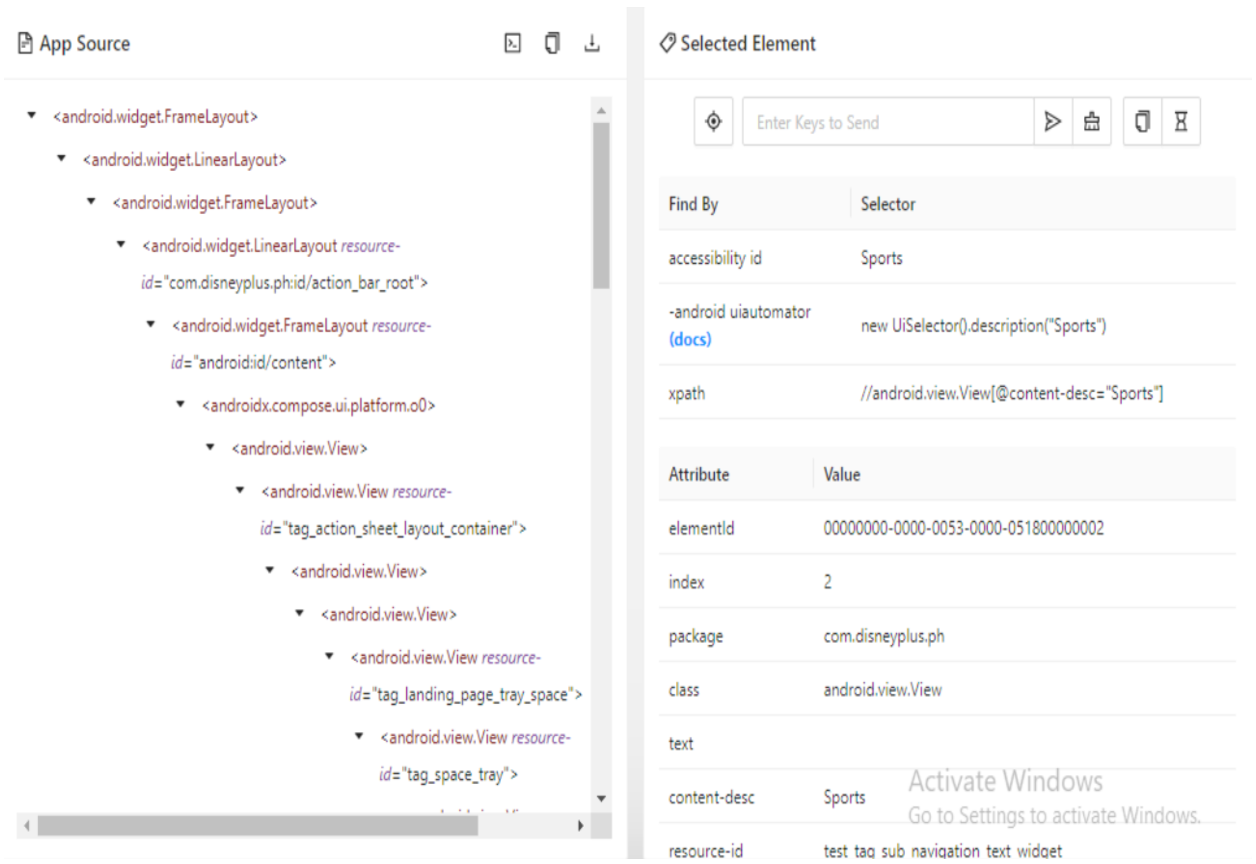
Click on Elements:

Click on any element on the screen. The inspector will display its properties (like resource-id, class, text, content-desc, etc.) in the right panel.

Copy Element Properties:

You can copy these properties to use in your test scripts. For example, you might use resource-id or content-desc to locate elements in your tests.





App Source

```
<android.widget.FrameLayout>
  <android.widget.LinearLayout>
    <android.widget.FrameLayout>
      <android.widget.LinearLayout resource-id="com.disneyplus.ph:id/action_bar_root">
        <android.widget.FrameLayout resource-id="android:id/content">
          <androidx.compose.ui.platform.o0>
            <android.view.View>
              <android.view.View resource-id="tag_action_sheet_layout_container">
                <android.view.View>
                  <android.view.View>
                    <android.view.View resource-id="tag_landing_page_tray_space">
                      <android.view.View resource-id="tag_space_tray">
```

Selected Element

Enter Keys to Send

Find By	Selector
accessibility id	Sports
-android uiautomator (docs)	new UiSelector().description("Sports")
xpath	//android.view.View[@content-desc="Sports"]

Attribute	Value
elementId	00000000-0000-0053-0000-051800000002
index	2
package	com.disneyplus.ph
class	android.view.View
text	
content-desc	Sports
resource-id	test_tag_sub_navigation_text_widget

THANK YOU