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

---

1. What is UIAutomatorViewer?

   UIAutomatorViewer is a tool provided by Google that helps you inspect the user interface (UI) of Android apps. It lets you see the layout of your app's UI elements and their properties, making it easier to write automated tests.

   Here's a simple way to understand it:

   - Think of UIAutomatorViewer as a magnifying glass for your app's UI. It shows you the structure of the app's screens and the details of each UI element.

   - It captures screenshots of your app and highlights all the elements, such as buttons, text fields, and images.

   - You can click on any element to see its properties, like its ID, text, and position on the screen.

   - This tool is especially useful for testers and developers who want to understand how different elements are laid out and interact with each other.

   Example: Imagine you have a mobile app with a login screen. You want to automate the process of entering a username and password and then clicking the login button. You open UIAutomatorViewer, take a screenshot of the login screen, and see that the username field has an ID of "usernameField," the password field has an ID of "passwordField," and the login button has an ID of "loginButton." This helps you write a test script that interacts with these elements by their IDs.

2. Understanding the UI Hierarchy

UI Hierarchy refers to the arrangement of UI elements in a structured way, similar to a family tree:

- At the top, you have the root element, which could be the main container of the app screen.

- Below the root, there are child elements, such as buttons, text fields, and images.

- Each child element can have its own children, creating a nested structure.

- Understanding this hierarchy is important because it helps you locate and interact with specific elements in your tests.

Example: Think of the UI hierarchy like a tree. The main screen of your app is the root. Under this root, you have branches like a header section, a main content section, and a footer section. Each branch can have its own smaller branches. For example, the main content section might have a form with text fields and buttons. Understanding this hierarchy helps you find the path to specific elements, like finding a leaf on a tree.

3. Understanding Desired Capabilities

Desired Capabilities are a set of key-value pairs used to configure your Appium test environment. They tell Appium what kind of device and app you want to test. Here's a simple breakdown:

- Device Name: Specifies the name of the device or emulator you're using (e.g., "Pixel_3").

- Platform Name: Indicates the operating system, like Android or iOS.

- App Package: The unique identifier for your app (e.g., "com.example.myapp").

- App Activity: The entry point of your app (e.g., "com.example.myapp.MainActivity").

- Automation Name: The framework used for automation (e.g., "UiAutomator2" for Android).

- These capabilities ensure that Appium sets up the test environment correctly and interacts with the right device and app.

Example: When you set up your Appium test, you need to tell it what device and app to use. You create a list of settings called "desired capabilities." For example:

Device Name: "Pixel_4"

Platform Name: "Android"

App Package: "com.example.myapp"

App Activity: "com.example.myapp.MainActivity"

Automation Name: "UiAutomator2"

These settings ensure that Appium uses the correct device and app for testing.

4. Inspecting Objects in Android Apps

Inspecting Objects means examining the UI elements in your app to understand their properties and how you can interact with them:

- Use tools like UIAutomatorViewer to capture screenshots of your app.

- Click on different elements in the screenshot to see their details, such as ID, class, text, and position.

- These details help you write accurate test scripts, as you need to know how to identify and interact with each element.

Example: You open your app and use UIAutomatorViewer to take a screenshot. You click on a button in the screenshot, and UIAutomatorViewer shows you that the button has an ID of "submitButton." Now, when you write your test script, you know to use this ID to interact with the button, like telling the script, "Click the button with the ID 'submitButton.'"

5. Deep Dive into Object Identification Strategies

Object Identification Strategies are methods used to locate and interact with UI elements in your app. Here are some common strategies:

- ID: The most straightforward way is to use the element's unique ID (e.g., `driver.findElementById("com.example.myapp:id/button")`).

- Class Name: Sometimes you can use the class name of the element, especially if IDs are not available.

- XPath: A powerful way to navigate through the UI hierarchy using a path expression (e.g., `driver.findElementByXPath("//button[@text='Login']")`).

- Accessibility ID: Useful for elements with accessibility labels (e.g., `driver.findElementByAccessibilityId("login_button")`).

- Text: You can locate elements based on the text they display (e.g., `driver.findElementByLinkText("Login")`).

- Each strategy has its pros and cons, and the choice depends on the structure of your app and the uniqueness of the element properties.

Example: You want to interact with a login button in your app. There are different ways to find this button:

ID: If the button has a unique ID, you use it (e.g., "submitButton").

Class Name: If IDs are not available, use the class name (e.g., "android.widget.Button").

XPath: If you need to navigate through the UI hierarchy, use XPath (e.g., "//button[@text='Login']").

Accessibility ID: If the button has an accessibility label (e.g., "login_button").

Text: If you want to find it by the text it displays (e.g., "Login").

6. Inspecting Elements in Android Apps

Inspecting Elements involves using tools and strategies to identify and understand the properties of UI elements in your app:

- Launch your app on a device or emulator and open UIAutomatorViewer.

- Capture a screenshot of the current screen in the app.

- Click on different elements in the screenshot to inspect their properties, such as ID, class, text, and layout.

- Take note of these properties for writing your test scripts.

- For example, if you want to automate a login process, you'll inspect the username and password fields and the login button to get their IDs or other unique properties.

- By understanding these properties, you can accurately interact with the elements in your automated tests, ensuring they perform the desired actions.

Example: You want to automate filling out a form in your app. You use UIAutomatorViewer to inspect the elements:

Username Field: ID is "usernameField"

Password Field: ID is "passwordField"

Submit Button: ID is "submitButton"

When writing your test script, you can use these IDs to interact with each element. For example, you tell the script, "Enter 'testUser' in 'usernameField', enter 'testPass' in 'passwordField', and click 'submitButton'."

By these, you can effectively use UIAutomatorViewer, understand the UI hierarchy, configure desired capabilities, inspect objects, and employ various strategies to identify and interact with elements in your Android app. This makes it easier to write reliable and efficient automated tests.

THANK YOU