Get Started with Android Testing in AIQ

Setup

Configure Environment Variables

- Set JAVA_HOME environment variable.
- Add JAVA_HOME/bin to PATH environment variable.

Install and Setup Android Studio

- Set ANDROID_HOME environment variable.
- Add ANDROID_HOME/platform-tools and ANDROID_HOME/tools to PATH environment variable.
- Download and install Android SDK Platforms.

Create Virtual Device (Android Emulator)

1. Start Android Studio
2. Click Configure
3. Click AVD Manager
4. Click Create Virtual Device
5. Select any Profile
6. Click Next

7. Select Any System Image
8. Click Next
    1. If button is greyed out, download any System Image
9. Provide Name
10. Click Finish

Configure your Appium Server Emulator or physical Android phone file

The Appium Server configuration file must be updated for your specific environment and Application. This is a JSON file that contains an array of objects. Each object corrresponds to a machine which has Appium running and the Emulator installed or physical Android phone connected. The required keys and values of each object are:

```
{
  “serverURL”: “http:/example.com:4723/wd/hub, URL in which Appium runs”,
  “proxyURL”: “http://example.com:9090, optional, URL in which Test Designer runs”,
  “capabilities”: {
    “platformVersion”: “9, Appium seems to automatically detect the version”,
    “udid”: “some identifier, must be in list of adb devices command”,
    “systemPort”: “8200 - 8299, to automate multiple Emulators or physical Android phones”,
    “deviceName”: “Android 1, can be whatever name”
  },
  “app”: “Absolute path to where .apk file is in local file system or URL of .apk file, needed if value for appPackage is empty and value for appActivity is empty”,
  “appPackage”: “Package of AUT, needed if value for app is empty, otherwise is optional if value for app is not empty”,
  “appActivity”: “Main Activity of AUT, needed if value for app is empty, otherwise is optional if value for app is not empty”
}
```

Download the **android-emulator-physical.json** file and open it in an editor.

Download here: android-emulator-physical.json

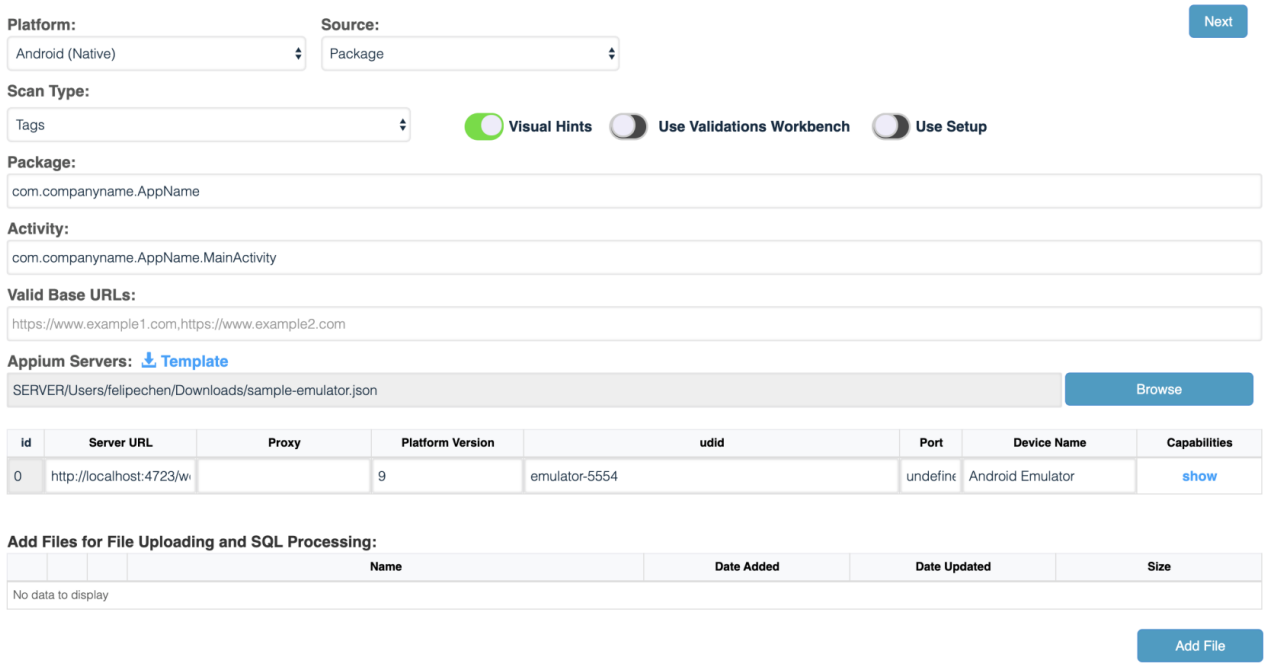
1. Replace the **udid** value in the template with the **Identifier number** of the Emulator or physical Android phone after running adb devices command.
2. Update the **appPackage** to match your Application's Package.
3. Update the **appActivity** to match your Application's Main Activity.

Create Android AISG

1. Open **Terminal**
2. Enter **appium** and press **Enter**

```
[Amanda-MacBook-Pro:aiq_distribution amandaburma$ appium
[Appium] Welcome to Appium v1.16.0
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

3. Start **AIQ** services if not already running
4. Log into **AIQ** and go to **AI Scripting** > **Create AISG**
5. Click **Mobile**
   If Mobile is disabled, request an updated AIQ license
6. Select **Android** in the **Platform** box
7. Select **Package** in the **Source** box
8. Select **Tags** in the **Scan Type** box (SmartTags not yet supported for Mobile)
9. Confirm **Visual Hints** are enabled
10. Click **Browse** in the **Appium Servers** box.
11. Select the Appium Server Simulator description file created earlier in the setup



12. Click **Next**
13. Proceed with the AISG creation
    1. Enter a name in **Execution Name** box
    2. Click **Load a Dataset from CSV file** and check the Datasets in the list or check Default and enter variables and values

3. Click **Create**

14. AISG starts the Application and begins discovering its elements
15. As Inputs Found updates, map any Custom Actions until the desired end point is reached in the Application
16. If proxyURL is not empty in Appium Server configuration file, Appvance certificate installed in Emulator or physical Android phone, Application is compiled to use Appvance certificate and proxy settings of Simulator or physical iPhone configured to point to value of proxyURL
    1. Click Requests tab to view captured Requests performed during AISG
    2. Download Requests Report
17. If Validations Workbench file is supplied, click Validations tab to view successful and failed Validations
18. Download scripts in JS format (DS not yet supported for Mobile) to run in JS Edit & Play or in a Scenario

---

Notes

- To run tests in Emulators, these must be already running since Appium won't start them up. If capability "avd": "name of Emulator" is provided, Appium will start only this Emulator.
- If capability "app": "absolute path or URL of .apk file" is provided, Appium will try to install Application if not installed.
- After connecting physical Android phone to computer, Trust this computer dialog may show, click Accept button and checkbox to remember setting.
- It is normal if Appium logs show many entries with ECONNRESET error.
- If Appium logs show ECONNREFUSED or socket hangup errors, remove AUT from memory, shutdown Emulator or disconnect physical Android phone, restart Appium, restart Emulator or reconnect physical Android phone.

Last updated: June 26, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

---

Revision #7
Created 2026-07-31 15:08:53 UTC by Admin
Updated 2026-11-17 21:58:52 UTC by AI_Mananger