Here's a step-by-step guide for configuring Jenkins to run Appium tests using a Freestyle project:

**Prerequisites**

1. **Jenkins Setup**: Ensure Jenkins is installed and accessible.
2. **Appium Setup**: Install Appium Server (npm install -g appium)
3. **Test Environment**:
   * Install Android SDK, configure environment variables (ANDROID\_HOME).
   * Ensure Appium test scripts are available.
   * Add Java to your system path (JAVA\_HOME).

**Step 1: Install Required Plugins**

* Open Jenkins Dashboard.
* Go to **Manage Jenkins > Plugin Manager**.
* Install these plugins:
  + **Maven Integration Plugin** (if using Maven)
  + **Test Results Analyzer Plugin** (for report visualization)
  + **HTML Publisher**

**Step 2: Create Freestyle Project**

1. Navigate to Jenkins Dashboard.
2. Click on **New Item**.
3. Enter a project name and select **Freestyle project**.
4. Click **OK**.

**Step 3: Configure Source Code Management (Optional)**

* Under **Source Code Management**, select **Git**.
* Provide the repository URL and credentials if needed.

**Step 4: Set Build Environment**

1. Go to the **Build Environment** section.
2. Select:
   * **Provide Node & npm bin/ folder to PATH** (if using Appium with Node.js).
   * **Set environment variables** (add custom environment variables if needed).

**Step 5: Configure Build Step**

1. Click **Add build step > Execute Shell**.
2. Enter the commands to run Appium and your test scripts:

bash

CopyEdit

# Start Appium Server

appium &

# Run Appium Tests using Maven or directly with Gradle

mvn clean test

# Kill Appium process

kill $(ps aux | grep 'appium' | awk '{print $2}')

**Step 6: Post-build Actions**

* Add **Publish JUnit test result report**.
* Enter the path to your test results (target/surefire-reports/\*.xml).

**Step 7: Save and Build**

1. Click **Save**.
2. Click **Build Now** to trigger a build.
3. Check the build logs to ensure the Appium server started and tests ran successfully.

**Additional Tips**

* For advanced configuration:
  + Use environment variables for Android paths.
  + Add stages for cleaning the workspace.
* For parallel test execution, consider converting the Freestyle project to a Pipeline.