

Prod Bug

By Sheetal Singh

Index:

1. What's new in Appium 2
2. Java-client 7 to Java-client 8
3. Complete Installation of Appium 2 Step by Step in Windows
 - a. Java, Maven
 - b. Node, NPM
 - c. Appium2, UIAutomator2, Appium Inspector
 - d. Android Studio
 - e. Create Mobile Devices
 - f. Sample Apk files
4. Different Locators – id, accessibilityId, xpath, className etc
5. Starting Appium 2 Server from Code
6. General Scrolling feature
7. Scrolling till element and click
8. Long press
9. Switching from Native App to Web view and back to Native App
10. Mobile Chrome Automation
11. Reporting
12. Mobile Automation Framework

#Links

<https://github.com/appium/appium>

<https://appium.io/docs/en/2.0/quickstart/install/>

#Appium 2.0 Installation Part

1. Install Java Maven

```
java -version
```

```
mvn -v
```

```
$ java -version
java version "18.0.2" 2022-07-19
Java(TM) SE Runtime Environment (build 18.0.2+9-61)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2+9-61, mixed mode, sharing)
```

```
$ mvn -v
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\Users\cmash\Downloads\apache-maven-3.8.6-bin\apache-maven-3.8.6
Java version: 18.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-18.0.2
Default locale: en_GB, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

Environment Variables:

JAVA_HOME C:\Program Files\Java\jdk-18.0.2

MAVEN_HOME C:\Users\cmash\Downloads\apache-maven-3.8.6-bin\apache-maven-3.8.6

2. Install Node/NPM

<https://nodejs.org/en/download>

Download msi file and check checkbox "install all necessary tools"

Installation path C:\Program Files\nodejs\

```
node -v
```

```
$ node -v
v18.16.0
```

```
$ npm -v
9.5.1
```

3. Install Appium 2.0

```
npm i -g appium@next
```

```
appium -v
```

```
$ appium -v
2.0.0-beta.66
```

To start Appium server

```
$ appium
[Appium] Welcome to Appium v2.0.0-beta.66
[Appium] Attempting to load driver uiautomator2...
[debug] [Appium] Requiring driver at C:\Users\cmash\.appium\node_modules\appium-uiautomator2-driver
reusing global emitter
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
[Appium] Available drivers:
[Appium]   - uiautomator2@2.24.0 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

<https://www.youtube.com/c/sheetalsingh23/videos>

#Note:

Note down following path which going to be used in code

```
String nodeExePath = "C:\\Program Files\\nodejs\\node.exe";
```

```
String nodeJSMainPath =
```

```
"C:\\Users\\cmash\\AppData\\Roaming\\npm\\node_modules\\appium\\build\\lib\\main.js";
```

There are lots of main.js, but we need to pick which is under appium > build > lib

4. Install Appium UIAutomator2

```
appium driver install uiautomator2
```

```
appium driver list
```

```
appium driver list --updates | To update installed drivers
```

```
$ appium driver install uiautomator2
- Installing 'uiautomator2' using NPM install spec 'appium-uiautomator2-driver'
✓ Installing 'uiautomator2' using NPM install spec 'appium-uiautomator2-driver'
i Driver uiautomator2@2.24.0 successfully installed
- automationName: UiAutomator2
- platformNames: ["Android"]
```

```
$ appium driver list
- Listing available drivers
✓ Listing available drivers
- uiautomator2@2.24.0 [installed (npm)]
- xcuitest [not installed]
- mac2 [not installed]
- espresso [not installed]
- safari [not installed]
- gecko [not installed]
- chromium [not installed]
```

5. Install Appium Doctor

```
npm install -g appium-doctor
```

```
appium-doctor --version
```

```
$ appium-doctor --version
1.16.2
```

```
appium-doctor --android
```

To check android setup related issues

```
$ appium-doctor --android
WARN AppiumDoctor [Deprecated] Please use appium-doctor installed with "npm install @appium/doctor --location=global"
info AppiumDoctor Appium Doctor v.1.16.2
info AppiumDoctor ### Diagnostic for necessary dependencies starting ###
info AppiumDoctor ✓ The Node.js binary was found at: C:\Program Files\nodejs\node.EXE
info AppiumDoctor ✓ Node version is 18.16.0
info AppiumDoctor ✓ ANDROID_HOME is set to: C:\Users\cmash\AppData\Local\Android\Sdk
info AppiumDoctor ✓ JAVA_HOME is set to: C:\Program Files\Java\jdk-18.0.2
```

6. Install Appium Inspector

Download Appium Inspector zip file from

<https://github.com/appium/appium-inspector/releases>

Unzip into Appium-Inspector folder and create shortcut and open Appium Inspector

Appium Inspector Web Application

<https://inspector.appiumpro.com/>

#Android Studio Installation Part

1. Install latest Android Studio via exe file

<https://developer.android.com/studio>

Give default values and click Next....sample MyApplication project will be created

2. Env variable setup

System variables	
Variable	Value
ANDROID_HOME	C:\Users\cmash\AppData\Local\Android\Sdk

Note: AppData is hidden folder in Windows

Directly add into path:

C:\Users\cmash\AppData\Local\Android\Sdk\build-tools
C:\Users\cmash\AppData\Local\Android\Sdk\platform-tools
C:\Users\cmash\AppData\Local\Android\Sdk\tools
C:\Users\cmash\AppData\Local\Android\Sdk\platforms

3. Verify Setup

adb devices or adb

4. Create Mobile Device

Open Android Studio Workspace

Go to > C:\Program Files\Android\Android Studio\bin

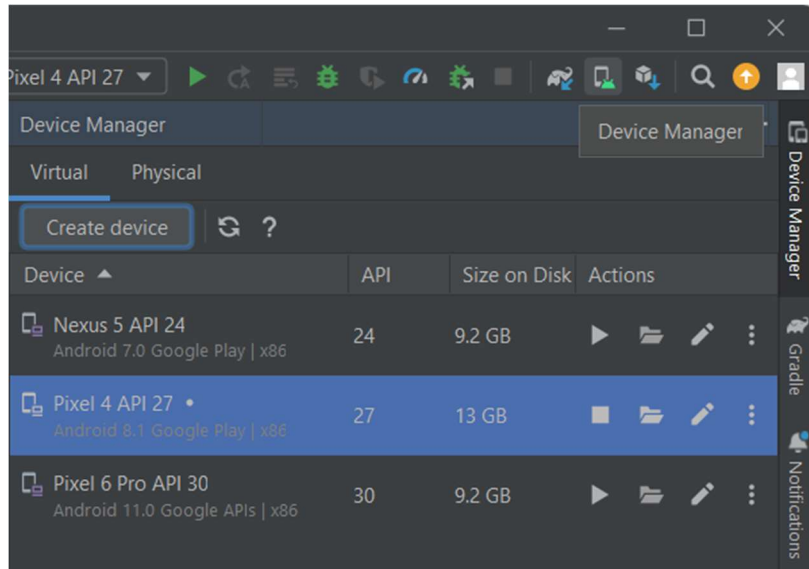
Double click studio.sh window batch file

Create studio.sh shortcut

Go to Device Manager (top right)

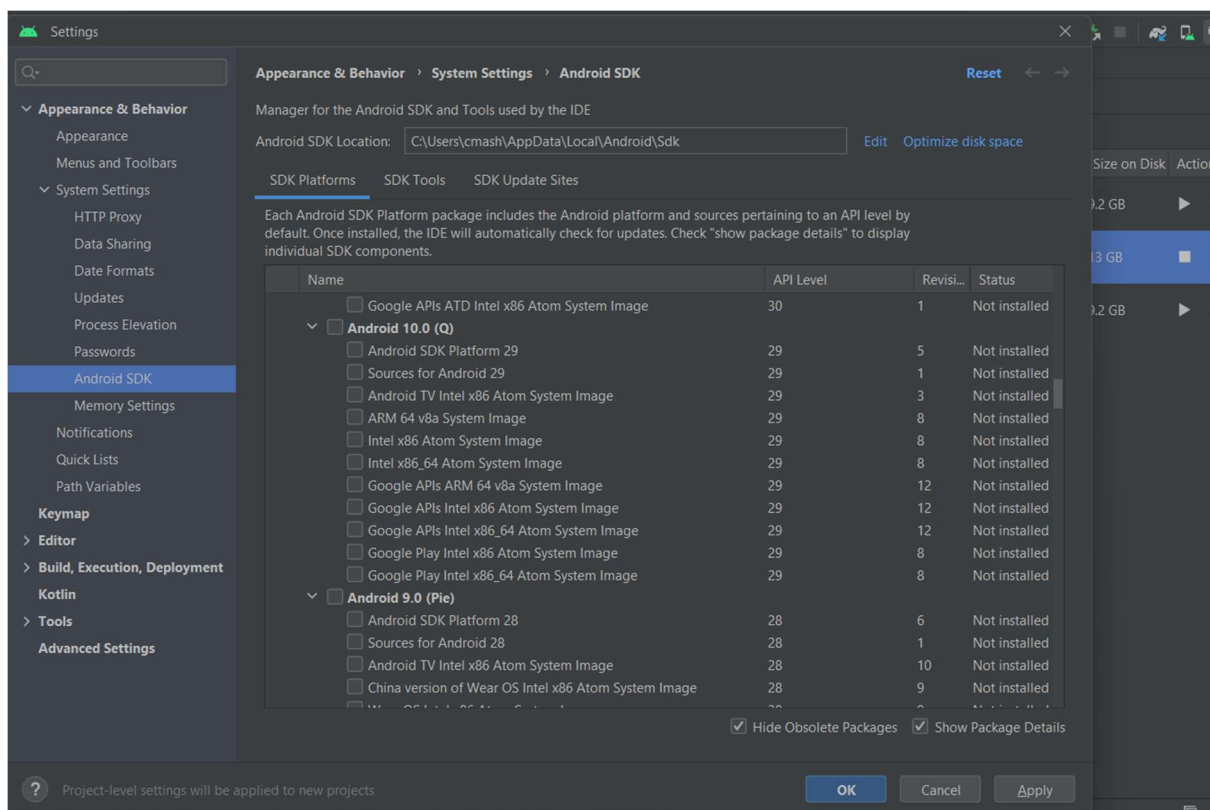
Select any device and create.... API number suggest Android

version



5. SDK Manager

In case some packages need to install in Emulators like Google Play Store.



6. Launch Emulator from created list

```
$ adb devices
List of devices attached
emulator-5554    device
```

7. Download Sample Apk Files

For installation, simply Drag and Drop apk into Emulator

App1:

ApiDemos-debug.apk

<https://github.com/appium-boneyard/sample-code/blob/master/sample-code/apps/ApiDemos/bin/ApiDemos-debug.apk>

App2:

Android-MyDemoAppRN.1.3.0.build-244.apk | ReactJS App

<https://github.com/saucelabs/my-demo-app-rn/releases>

App3:

Android-NativeDemoApp-0.4.0.apk | WebDriver IO Apk

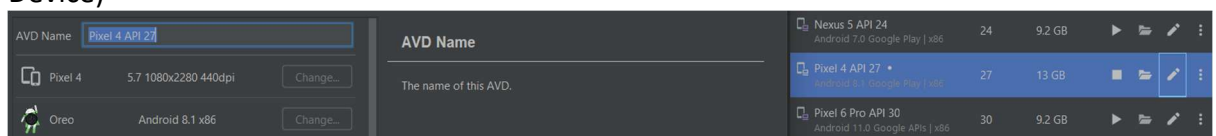
<https://github.com/webdriverio/native-demo-app/releases>

#Important Steps to Start Appium Session

1. Start Appium 2.0 Server first (cmd is "appium")
2. Start Appium Inspector after Appium started properly
3. Start Android Emulator
4. Add Desired capabilities in Appium Inspector and "Start Session"

#UIAutomator2Options

1. platformName = Android
2. platformVersion = 8.1, Emulator settings will show platform version (Edit Emulator Device)



Another way to get "platformVersion" capabilities

adb shell getprop ro.build.version.release

```
$ adb shell getprop ro.build.version.release
8.1.0
```

- automationName = UIAutomator2
- deviceName: emulator-5554, cmd: "adb devices"

```
$ adb devices
List of devices attached
emulator-5554    device
```

- appPackage/appActivity:

adb shell dumpsys window windows | grep mFocusedApp

```
$ adb shell dumpsys window windows | grep mFocusedApp
mFocusedApp=AppWindowToken{74af66e token=Token{5351bca ActivityRecord{be80b35
u0 io.appium.android.apis/.ApiDemos t313}}}
```

#Appium Inspector Desktop Application

[Appium Server](#) [Select Cloud Providers](#)

Remote Host127.0.0.1

Remote Port4723

Remote Path/

☐ SSL

> Advanced Settings

Desired Capabilities

Saved Capability Sets 0

Attach to Session...

appium:platformName	text	Android	✕
appium:platformVersion	text	8.1	✕
appium:automationName	text	UIAutomator2	✕
appium:deviceName	text	emulator-5554	✕
appium:appPackage	text	io.appium.android.apis	✕
appium:appActivity	text	.ApiDemos	✕

☒ Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{
  "appium:platformName": "Android",
  "appium:platformVersion": "8.1",
  "appium:automationName": "UIAutomator2",
  "appium:deviceName": "emulator-5554",
  "appium:appPackage": "io.appium.android.apis",
  "appium:appActivity": ".ApiDemos"
}
```

Note: Appium Inspector Web Application

<https://inspector.appiumpro.com/>

<https://www.youtube.com/c/sheetalsingh23/videos>

Appium Demo ApiDemos-debug.apk

```
{  
  "appium:platformName": "Android",  
  "appium:platformVersion": "8.1",  
  
  "appium:automationName": "UIAutomator2",  
  "appium:deviceName": "emulator-5554",  
  
  "appium:appPackage": "io.appium.android.apis",  
  "appium:appActivity": ".ApiDemos"  
}
```

Saucelabs Demo Android-MyDemoAppRN.1.3.0.build-244.apk

```
{  
  "appium:platformName": "Android",  
  "appium:platformVersion": "8.1",  
  "appium:automationName": "UIAutomator2",  
  "appium:deviceName": "emulator-5554",  
  "appium:appPackage": "com.saucelabs.mydemoapp.rn",  
  "appium:appActivity": ".MainActivity"  
}
```


#Java-client v7 vs v8

<https://github.com/appium/java-client/blob/master/docs/v7-to-v8-migration-guide.md>

Java client 8 support Selenium 4 and strictly compliant to W3C

Duration class handles all time related things

1. MobileElement has been removed and now we have to use WebElement
2. DesiredCapability removed and now we have UiAutomator2Options
3. "java-client" dependency already have "selenium-java".
Just we need to add "selenium-support" for ExpectedCondition and WebDriverWait
4. TouchActions Depreciated....Action introduced
5. MobileBy replaced with AppiumBy
6. LaunchApp, CloseApp, ResetApp Deprecated
7. java.time.Duration time to be used
8. Use sendKeys rather setValue
9. Old: `http://0.0.0.0:4723/wd/hub`
New: `http://0.0.0.0:4723/`

#Locators Strategies

Appium Inspector Locators	Java Code to Be Used
accessibility id	<code>driver.findElement(AppiumBy.accessibilityId("abc"))</code>
content-desc	<code>driver.findElement(AppiumBy.accessibilityId("abc"))</code>
resource-id	<code>driver.findElement(AppiumBy.id("abc"))</code>
xpath	<code>driver.findElement(AppiumBy.xpath("//abc"))</code>

Note:

AppiumBy is Child of By and hence can also be used

```
import org.openqa.selenium.By;
```

```
import io.appium.java_client.AppiumBy;
```

#XPath Types:

```
//android.widget.TextView[@content-desc="App"]
```

```
//android.widget.TextView[@resource-id="android:id/text1"]
```

```
//android.widget.TextView[@index=3]
```

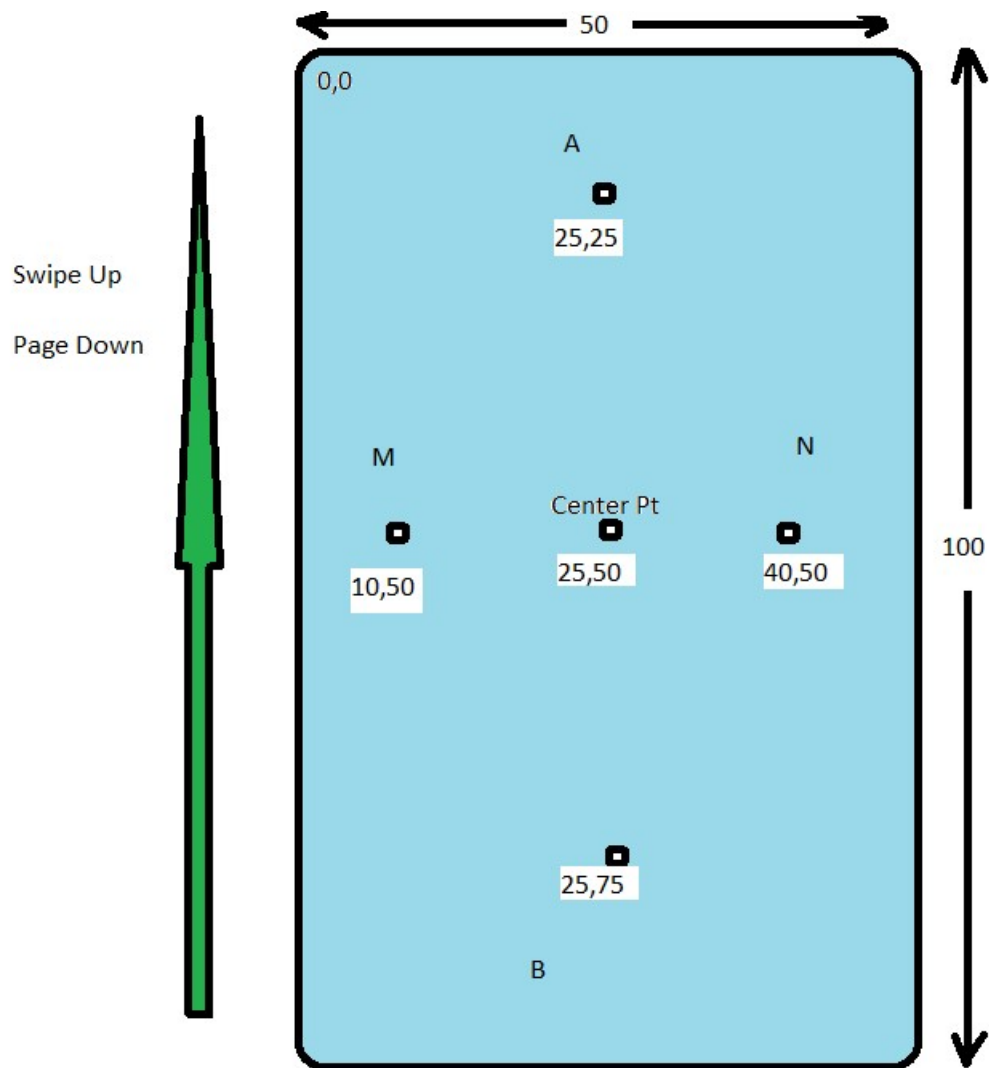
```
//*[contains(@text,'Read')]
```

```
//android.widget.TextView[@text="Assets"]
```

```
//*[@resource-id='android:id/text1' or @text='Media']
```

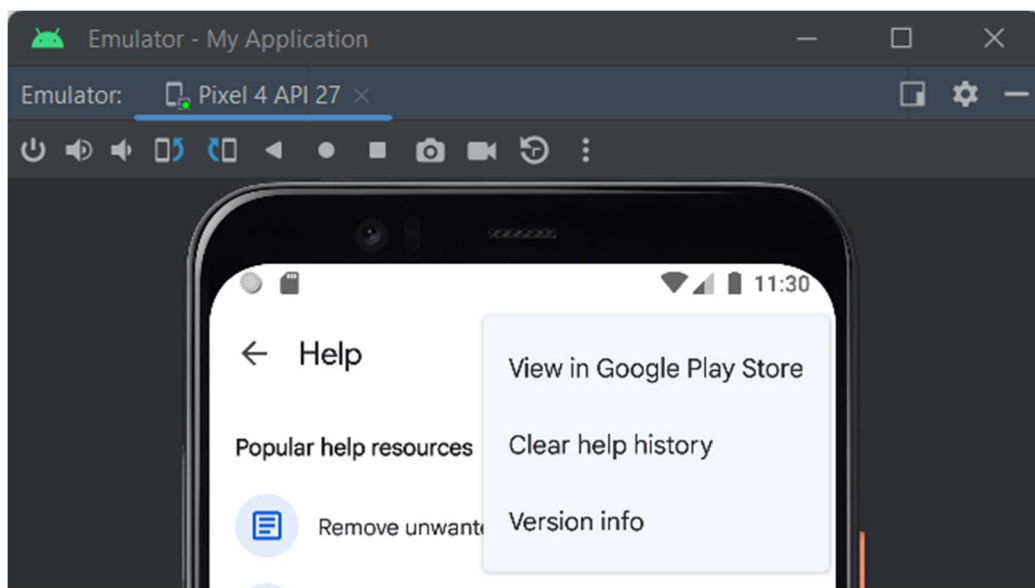
```
//*[@resource-id='android:id/text1' and @text='Media']
```

#Swipe function understanding



Appium Chrome Browser Automation:

1. Download Katalon Recorder on Desktop browser
https://chrome.google.com/webstore/detail/katalon-recorder-selenium/ljdobmomdgdlniojadhoplhkpialdid?utm_source=chrome-ntp-icon
2. Download Chrome Driver based on Mobile emulator chrome version
<https://chromedriver.chromium.org/downloads>

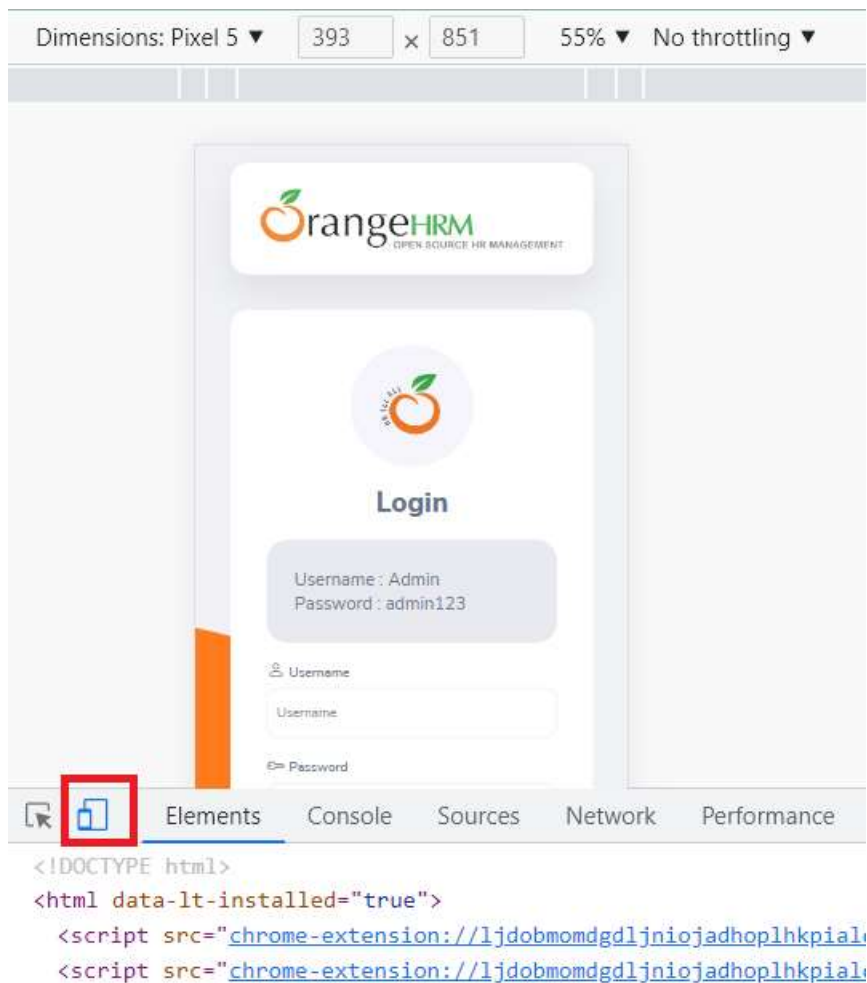


3. Save correct chrome driver in
C:\Users\cmash\.appium\node_modules\appium-uiautomator2-driver\node_modules\appium-chromedriver\chromedriver\win\

You can also run following command when appium is already up and running

```
appium --allow-insecure chromedriver_autodownload
```

4. Demo website to be automated
<https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>
5. Locators can be directly checked from Desktop browser and mobile view can be taken from Chrome's Developers tool



6. Capabilities as per Appium 2

```
{
  "appium:platformName": "Android",
  "appium:platformVersion": "8.1",
  "appium:automationName": "UIAutomator2",
  "appium:deviceName": "emulator-5554",
  "appium:withBrowserName": "Chrome"
}
```

```
UiAutomator2Options options = new UiAutomator2Options();
options
  .setPlatformName("Android")
  .setPlatformVersion("8.1")
  .setAutomationName("UiAutomator2")
  .setDeviceName("emulator-5554")
  .noReset()
  .withBrowserName("Chrome");
```

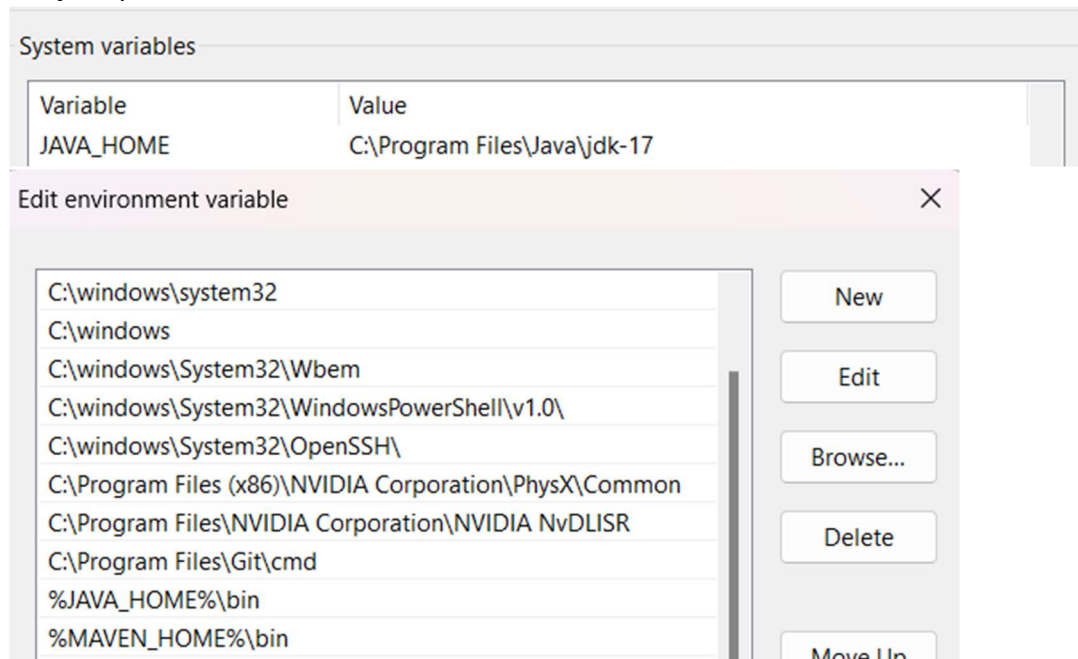
Running Tests on Jenkin

Precondition:

1. Install Java 11 or 17
2. Go to <https://www.oracle.com/java/technologies/downloads/#java17>

Download x64 MSI Installer

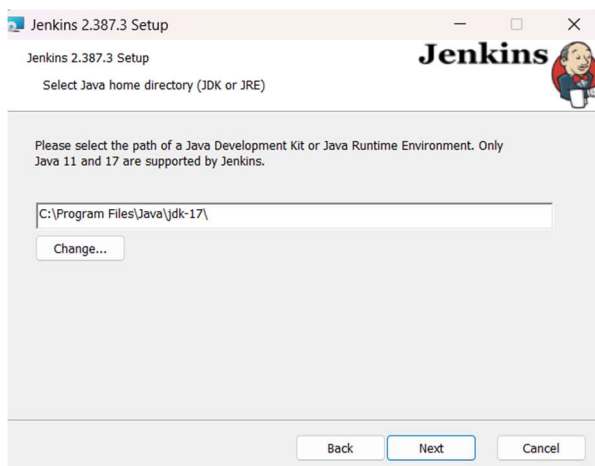
3. Set java path



```
$ java -version
java version "17.0.7" 2023-04-18 LTS
Java(TM) SE Runtime Environment (build 17.0.7+8-LTS-224)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.7+8-LTS-224, mixed mode, sharing)
```

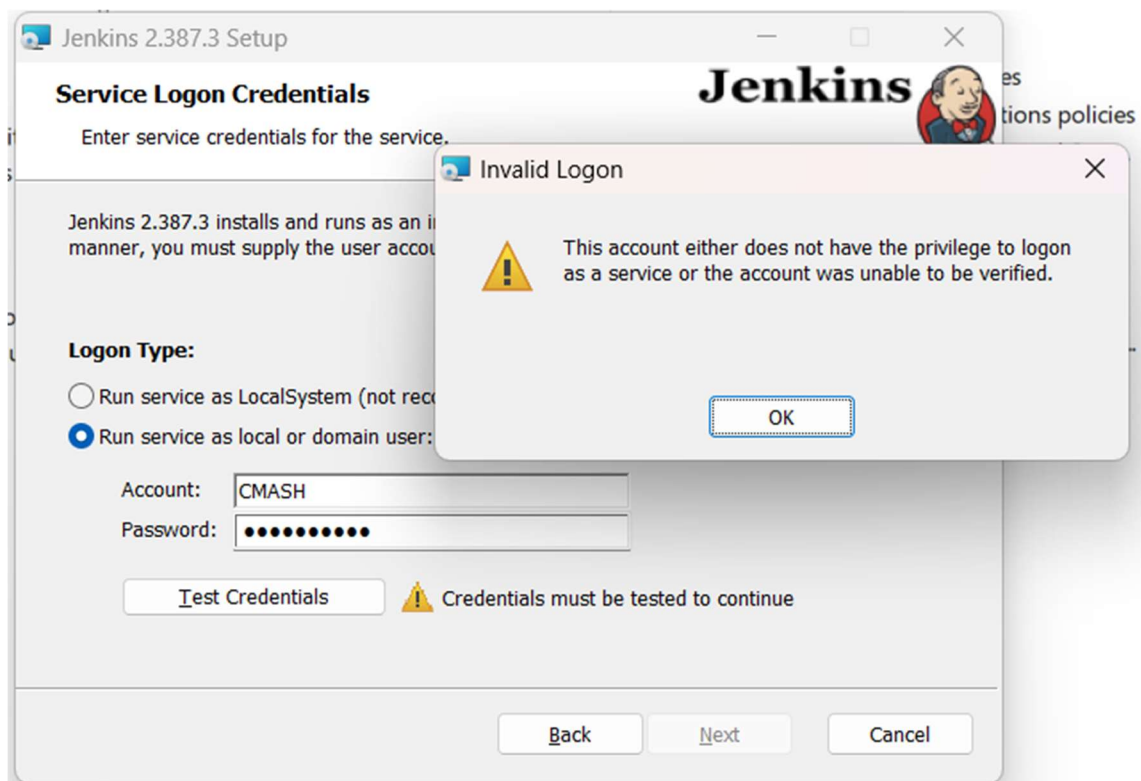
Note:

As of now Jenkin Support Java 11 and 17. So make sure we have proper Jdk



Jenkins Setup

1. Download Jenkin (Jenkins.msi)
<https://www.jenkins.io/download/>
<https://www.jenkins.io/doc/book/installing/windows/>
2. Most of the time user don't have privilege
<https://www.jenkins.io/doc/book/installing/windows/#invalid-service-logon-credentials>

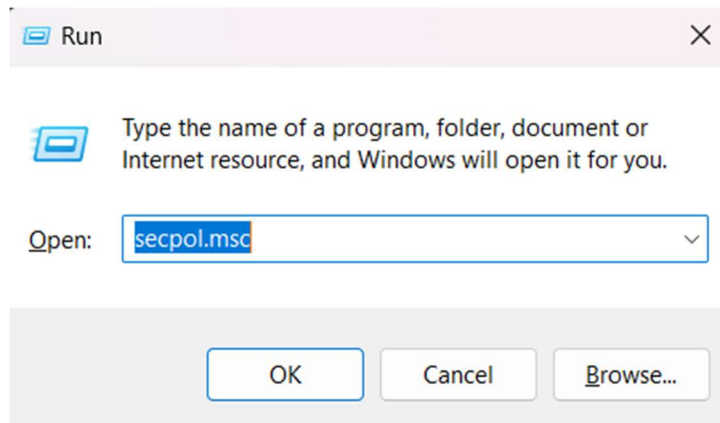


3. Install gpedit.msc
<https://www.itechtics.com/enable-gpedit-windows-10-home/>
 - a. Open CMD in Admin mode
 - b. Run following three commands one by one (refer above link)
FOR %F IN ("%SystemRoot%\servicing\Packages\Microsoft-Windows-GroupPolicy-ClientTools-Package~*.mum") DO (DISM /Online /NoRestart /Add-Package:"%F")

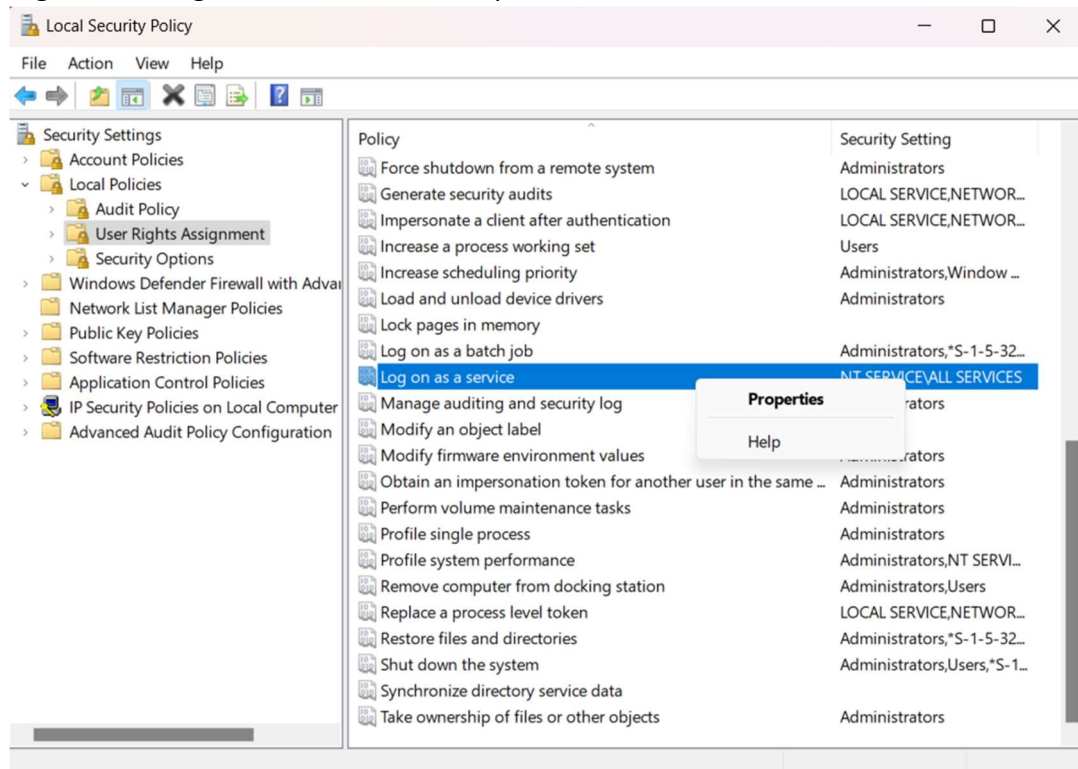
FOR %F IN ("%SystemRoot%\servicing\Packages\Microsoft-Windows-GroupPolicy-ClientExtensions-Package~*.mum") DO (DISM /Online /NoRestart /Add-Package:"%F")

gpedit.msc

4. Win + R



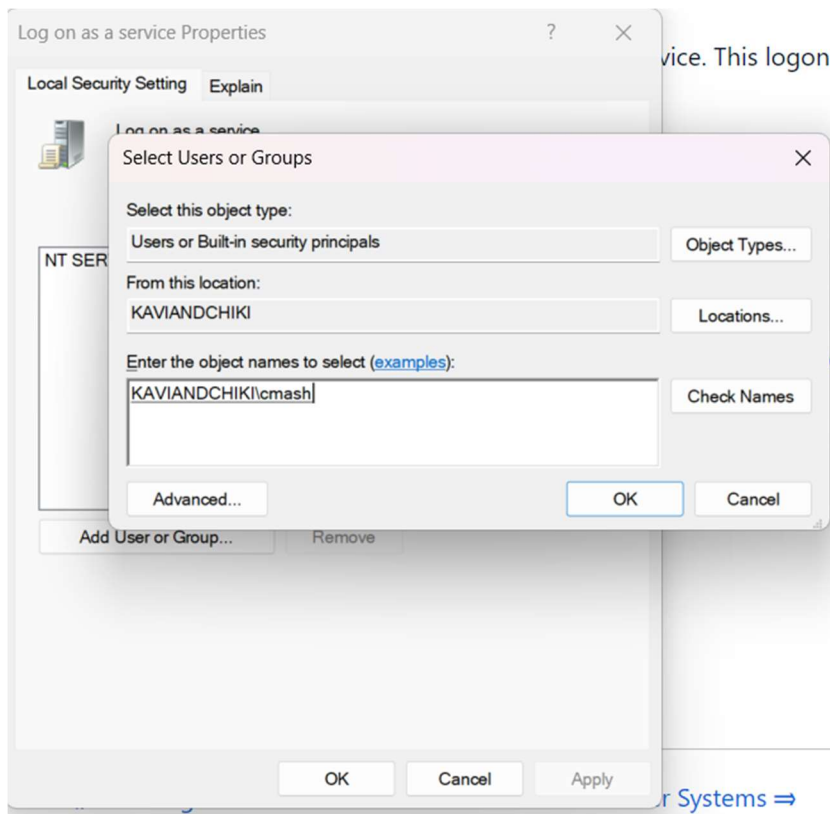
5. Right click "Log on as a Service" > Properties



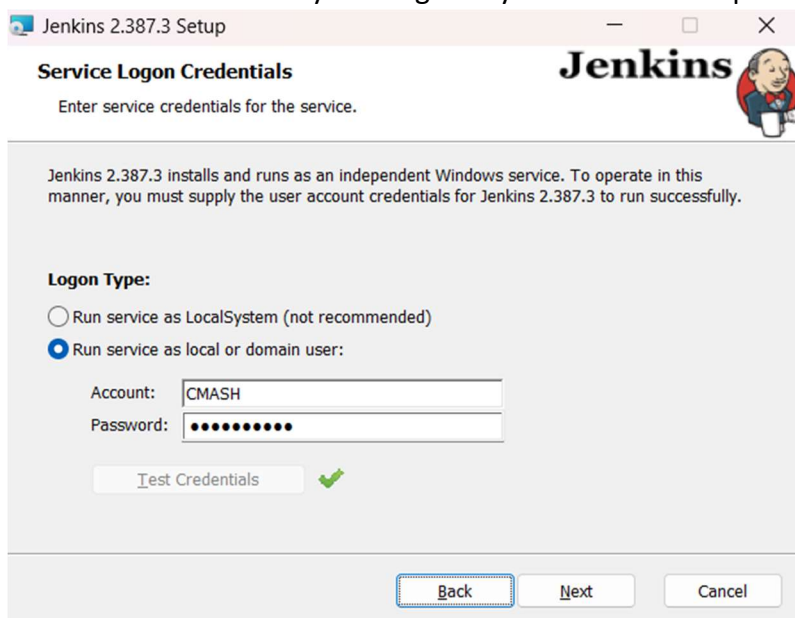
6. Add your username properly and click on "Check names"

To get username > open cmd and type > whoami

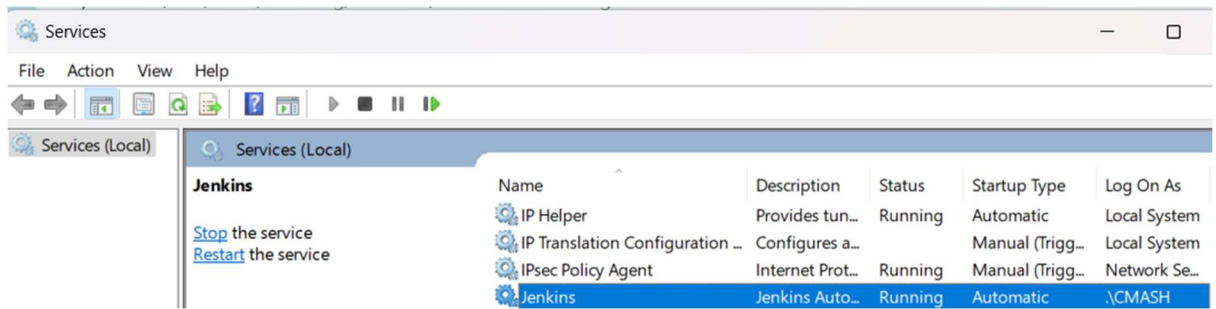
```
$ whoami
cmash
```

7. Once added successfully then again try with Jenkin Setup



8. After jenkins setup finish. Open Service and verify if Jenkins is running or not



9. Open <http://localhost:8080/login?from=%2F>

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

`C:\Users\cmash\AppData\Local\Jenkins\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

10. Go to mentioned location and paste given password
 11. Install recommended plugins or your own plugins

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	Credentials Binding ** SCM API ** Pipeline: API ** commons-lang3 v3.x Jenkins API
✓ Timestamper	Workspace Cleanup	Ant	Gradle	Timestamper ** Caffeine API ** Script Security ** JAXB ** SnakeYAML API ** Jackson 2 API ** commons-text API ** Pipeline: Supporting APIs ** Plugin Utilities API ** Font Awesome API ** Bootstrap 5 API
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

12. Enter user details and click Save and Continue (admin/M@123)

Create First Admin User

Username

admin

Password

.....

13. URL to open <http://localhost:8080/>

#Jenkin Job

1. Configure Maven in Jenkin

Go to <http://localhost:8080/manage/configureTools/>

Dashboard > Manage Jenkins > Global Tool Configuration

Maven

Maven installations ^ Edited

Maven installations

List of Maven installations on this system

Add Maven

Maven Name

M3

☒ Install automatically ?

Install from Apache

Version

3.8.6

Add Installer ▾

Save Apply


2. Install TestNG Plugin

3. Go to <http://localhost:8080/manage/pluginManager/updates/>

Search and install TestNG plugin “TestNG Results Plugin”

For more info <https://plugins.jenkins.io/testng-plugin/dependencies/>

4. Go to Dashboard > New Item > Create a Pipeline


 **Jenkins**


Dashboard > All >

Enter an item name

SeleniumPipeline

» A job already exists with the name 'SeleniumPipeline'

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, can be used for something other than software build.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agent and/or organizing complex activities that do not easily fit in free-style j

5. Select Github + Maven and it will generate script.

Update maven command to be run

Definition

Pipeline script

Script ?

1

try sample Pipeline...
try sample Pipeline...
Hello World
GitHub + Maven
Scripted Pipeline

Pipeline

Definition

Pipeline script

Script ?

```

1 pipeline {
2   agent any
3   tools {
4     maven "M3"
5   }
6   stages {
7     stage('Build') {
8       steps {
9         // Get some code from a GitHub repository
10        git 'https://github.com/sheetalrepo/AppiumTest.git'
11        bat "mvn clean test -DsuiteXmlFile=seleniumTestNG.xml"
12      }
13    }
14    post {
15      always {
16        junit 'target/surefire-reports/*.xml'
17        testNG(reportFilenamePattern: 'target/surefire-reports/testng-results.xml')
18      }
19    }
20  }
21 }
22

```

GitHub + Maven

☒ Use Groovy Sandbox ?

Pipeline Syntax

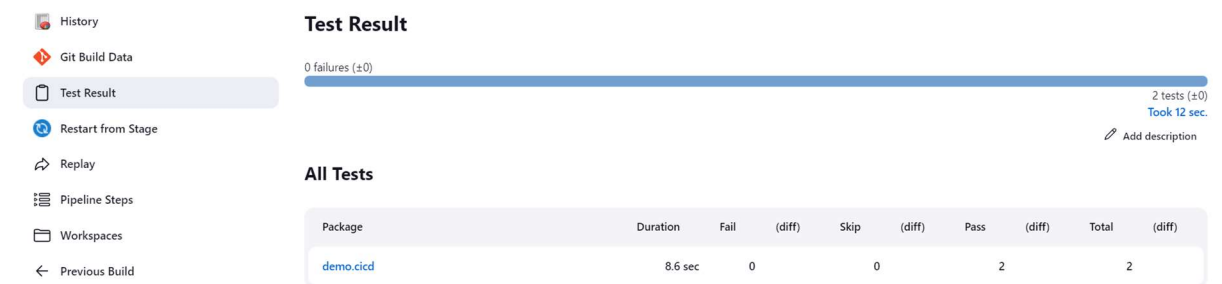
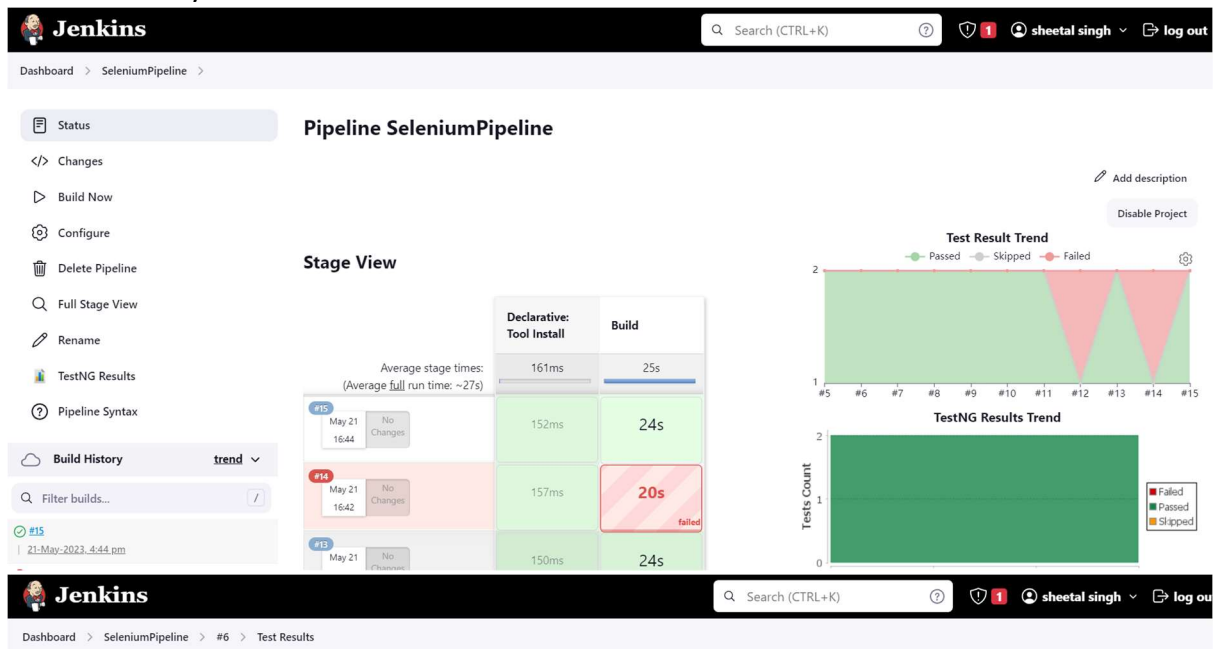
```

pipeline {
  agent any
  tools {
    maven "M3"
  }
  stages {
    stage('Build') {
      steps {
        git 'https://github.com/sheetalrepo/AppiumTest.git'
        bat "mvn clean test -DsuiteXmlFile=seleniumTestNG.xml"
      }
    }
    post {
      always {
        junit 'target/surefire-reports/*.xml'
        testNG(reportFilenamePattern: 'target/surefire-reports/testng-
results.xml')
      }
    }
  }
}

```

}}

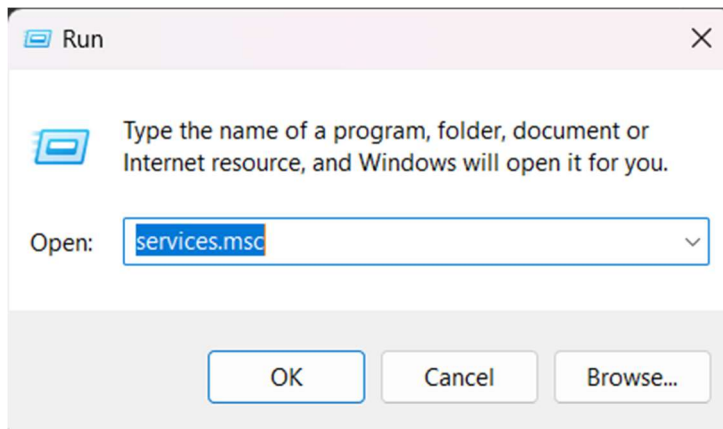
- Click on Build Now
- Report shows Test cases Trends over the period
Good for Analysis



#To make Chrome visible in Window machine when Selenium runs on Jenkin

(Not Working, Chrome still runs in background)

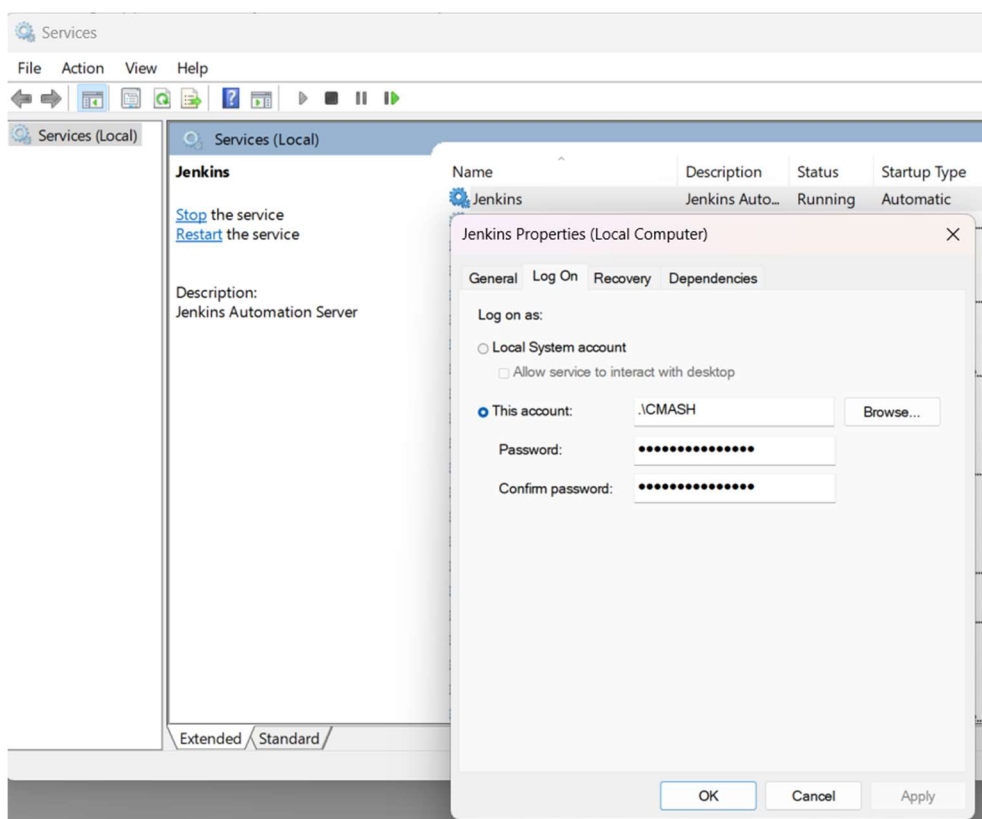
1. Press Win + R



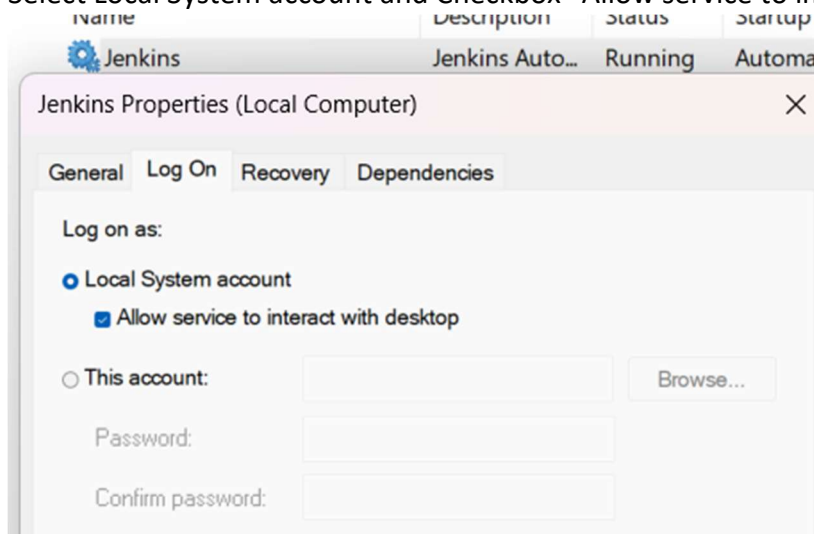
2. Locate Jenkin Service

Right click and open properties > Log On

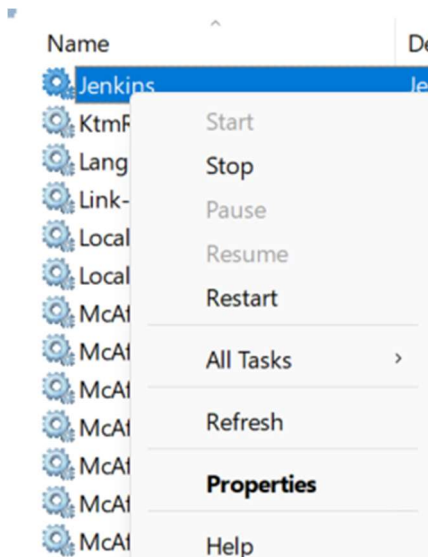
Default view



3. Select Local System account and Checkbox “Allow service to interact with Desktop”



4. Save and restart Jenkin Service



5. Remember all previous saved jobs will be lost
Again, verify whether saved changes are properly saved or not

#General Exceptions:

1. Appium Inspector not able to Start Session with following error

```
[debug] at AppiumDriver.cre
<-- POST /session 500 3276 ms - 716

[Appium] Received SIGINT - shutting down
[debug] There are no active sess
Waiting until the server is closed
Received server close event
```

Solution 1: Sometimes connection break with UIAutomator2, run following command and it should start working.

```
adb uninstall io.appium.uiautomator2.server.test
```

```
$ adb uninstall io.appium.uiautomator2.server.test
Success
```

Solution 2: Close everything and again start in proper sequence. Reinstall APK by Drag and Drop.

Appium Server > Appium Inspector > Emulator Device

Optional Desired Capability:

"appium:noReset": true | app will not be reinstalled if its already there

2. Receiver class io.appium.java_client.service.local.AppiumServiceBuilder does not define or inherit an implementation of the resolved method 'abstract void loadSystemProperties()' of abstract class org.openqa.selenium.remote.service.DriverService\$Builder.

Sol: javaclient version increased from 8.3.0 to 8.5.0