

Appium advanced

```
>git command to push code from eclipse
git init
git status
git add .
git status
git commit -m "first commit"
git remote add origin https://github.com/Aroli1994/AppiumBasicProject.git
git remote -v
git branch
git push origin main =>username & token
git config --global http.postBuffer
git config --global http.postBuffer 157286400
git push origin main =>username & token. =>successfully code pushed to
empty GitHub repository
```

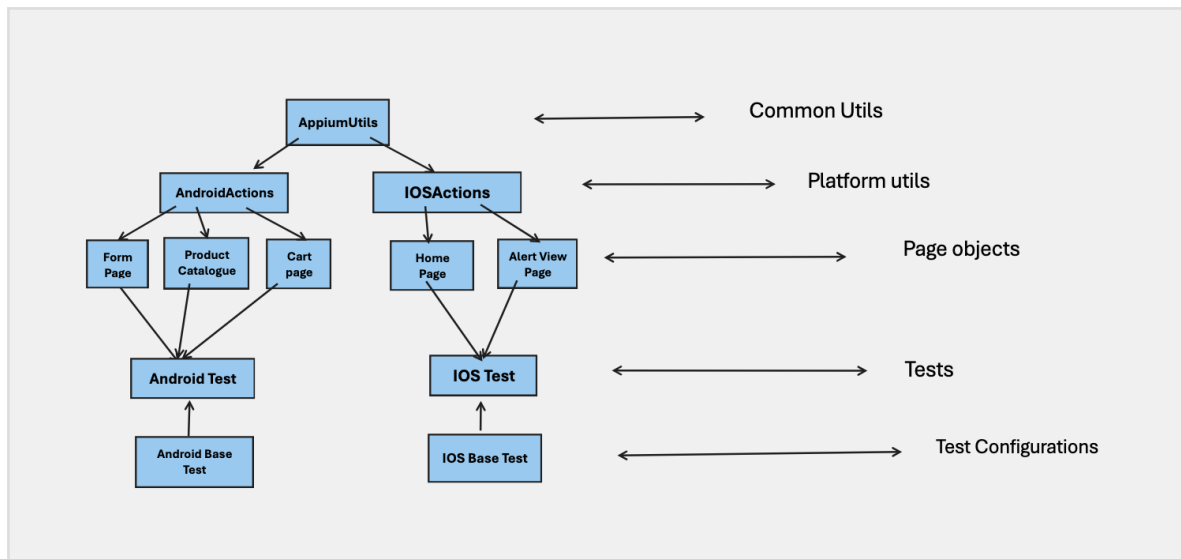
```
>To remove duplicate values in $PATH variables in MAC, execute below
commands
typeset -U path
echo $PATH | tr ':' '\n'
```

```
>qemu system aarch64(Android Emulator):
cd ~/Library/Android/sdk
./emulator/emulator -avd Vinay_Pixel8 -dns-server 8.8.8.8
```

>if we need to use testNG maven dependency greater than 6.0.0 version, then our java or jdk version should be minimum of 11.

>if we are using idk 1.8 version then testing maven dependency should not be greater than 6.0.0 version

Appium Framework design:

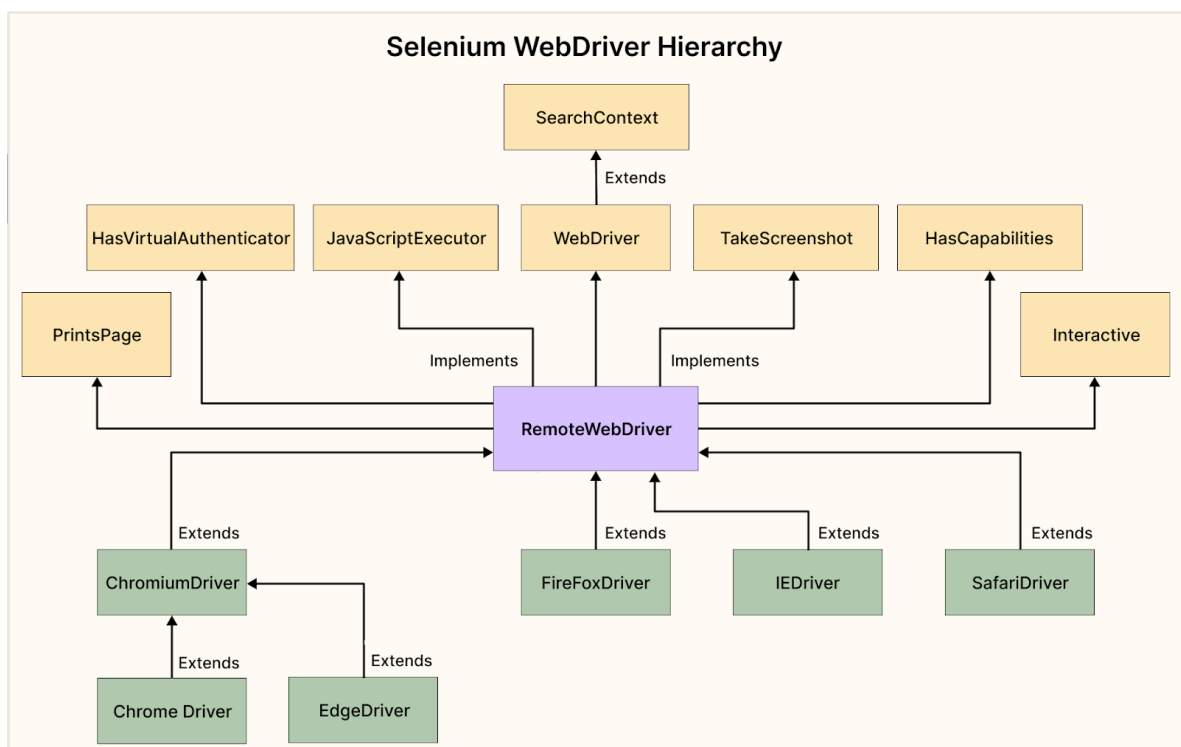


> Important ->

AppiumDriver(C)

IOSDriver(C) extends AppiumDriver

AndroidDriver(C) extends AppiumDriver



```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
```

```
<suite name="Suite">
```

```
<test thread-count="5" name="Test">
```

```
<classes>
```

```
<class name="org.rahulshettyacademy.eCommerce_tc_4_Hybrid"/>
```

```
<class name="org.rahulshettyacademy.IOSBasics"/>
```

```

    <class name="org.rahulshettyacademy.eCommerce_tc_1"/>
  </classes>
</test> <!-- Test -->
</suite> <!-- Suite -->

```

```

>Parse Json file  -> Json String      =>(Common-io dependency)
>Json String      -> HashMap          =>(Jackson-databind dependency)
>HashMap          -> testcase          =>(TestNG Data provider)

```

Appium(Android & IOS) Framework Development from Scratch:

- Understand the importance of Framework development
- Build Page object Factory pattern to drive the Element locators to the Tests
- Implement Android & IOS page object repositories for the respective Platform tests
- Build Actions utility classes for both Android & IOS Apps to store reusable Mobile Gestures
- Build Generic Utility called Appium Utils Class which have commonly used methods for Android & IOS
- Build Test Strategy on how tests should be written in isolation with setup & teardown methods
- Data driven testing from json payloads into Test using Jackson Mapper & TestNG Data Provider
- Implement BaseTest with Capabilities where all the Tests inherit it into their respective classes
- Methods to dynamically Start & Stop Appium Server before every Test
- Create Properties file to implement Global configuration values into the Framework
- Build TestNG.xml file to run all tests together with proper class, Suite, Test conversions inside the xml file
- Introduce Extent reports to get excellent HTML reports of Framework results
- Implement TestNG listeners to capture screenshots on test failures & report it back to Extent reports
- Implement TestNG groups to run selected tests based on tags set on Test level

>Information on how to run our project from maven commands

<https://maven.apache.org/surefire/maven-surefire-plugin/examples/testng.html>

```

<profile>
<id>Regression</id>
<build>

```

```

<pluginManagement>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.5.0</version>
<configuration>
<suiteXmlFiles>
<suiteXmlFile>testng.xml</suiteXmlFile>
</suiteXmlFiles>
</configuration>
</plugin>
</plugins>
</pluginManagement>
</build>
</profile>

```

```

mvn test -PRegression
mvn test -PSmoke

```

Here, -P stands for Profile

```

mvn test -PRegression -DipAddress=127.0.0.1

```

Here, -D stands for runtime parameters which we are planning to send through command

Below is using ternary operator to either initialise value of 'ipAddress' through runtime from command prompt OR from data.properties file from Eclipse

```

String ipAddress = System.getProperty("ipAddress") != null ?
System.getProperty("ipAddress") :
prop.getProperty("ipAddress");

```

>Jenkins

Download Jenkins (Generic Java package -> .war) from <https://www.jenkins.io/download/>

>To launch Jenkins

Move to location where we have placed the jenkins.war

```

java -jar jenkins.war -httpPort=9090

```

password to proceed to installation: dd9dceb12aa547639a804ed973c676ce

Location of initial password: /Users/krishnabros/.jenkins/secrets/

initialAdminPassword

<http://localhost:8080/>. => to launch Jenkins in google chrome and set up initially

Jenkins credentials:

admin
admin

Build Steps : Invoke top-level Maven targets

Goals: test -PRegression

But above parameter is hardcoded with 'Regression' instead of choosing parameter at run. So follow below one

General ->

👍 This project is parameterised

So configure Choice Parameters

Name: profile

Choices:

Regression

Smoke

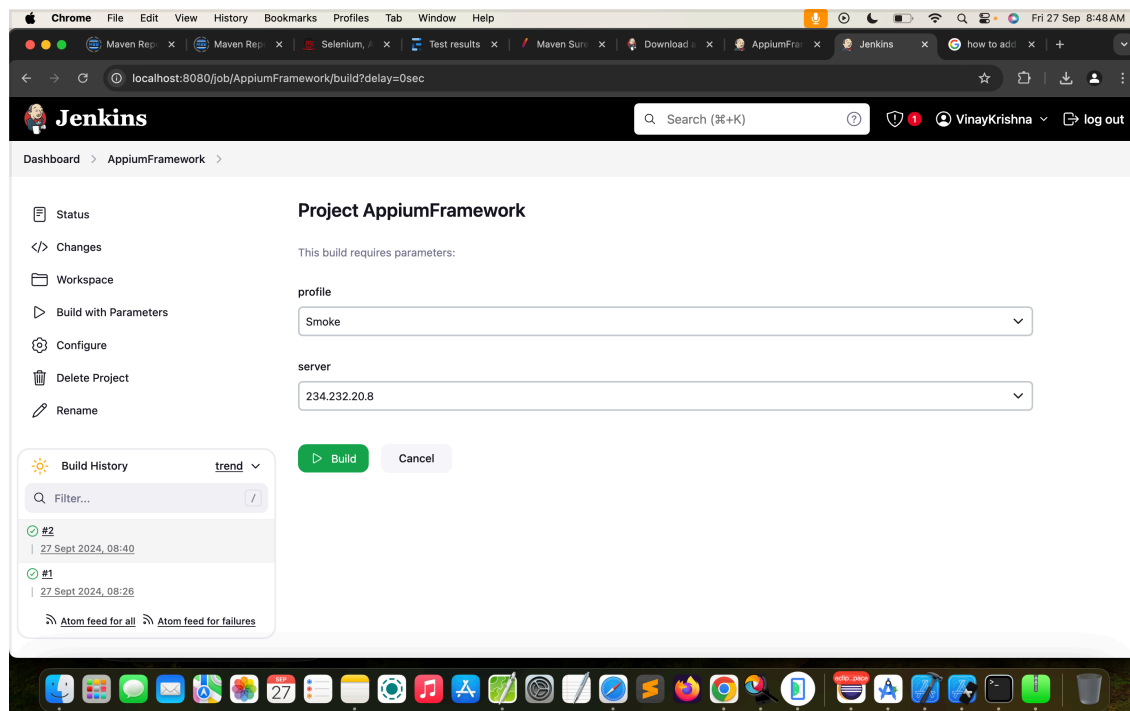
Build Steps : Invoke top-level Maven targets

Goals: test -P"\$profile"

Hence, Build button got changed to Build with Parameter & based on parameters we chose at runtime.

Goals: test -PRegression -DipAddress=127.0.0.1

Goals: test -P"\$profile" -DipAddress="\$server"



> How to schedule jenkins jobs in jenkins - Please refer below documentation which is really understandable

<https://www.cloudbees.com/blog/how-to-schedule-a-jenkins-job>

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

User: admin

Pass: admin

>To get appium interview questions & resume preparation, email to below id

mentor@rahulshettyacademy.com

>Cloud Infrastructure testing - Integrate Browser stack to Appium framework

Appium framework with Cloud Integration (Browser stack)

- Create Account in Browser stack
- Install Browser stack plugin to Eclipse
- Convert Project to Browser stack compatible
- Understand Yaml file and fill login and App details
- Run the tests
- Understand how to switch between local and cloud execution

>Different vendors who provides cloud infrastructure are
Browser stack

Lambda test
Saucelabs
Headspin
Perfecto
SmartBear Bitbar
Testsigma
Mobitru

>Test automation tool to test mobile apps

- Appium
- Espresso
- XCUITest
- Flutter

Browserstack credentials:

vinaykrishna_6Gqs1l

2WWpSGysshXizfHyvhn5

[bs://62bd2271c5e278bc777b850a1018362564eb99e5](https://62bd2271c5e278bc777b850a1018362564eb99e5)

```
<argLine>-javaagent:"${com.browserstack:browserstack-java-sdk:jar}"</argLine>
```

>To clean the maven project

```
mvn clean install -U
```

-U re-downloads all of the dependencies

>If we don't want to run in browser stack

```
BROWSERSTACK_AUTOMATION=false mvn test -PRegression
```