|  |  |
| --- | --- |
| ArcGIS Native Apps | Abstract  The ArcGIS\_Collector automation projects is designed and developed to automate the Sanity Test run for ArcGIS Collector Daily build. It’s a Maven Project using TestNG and Jenkins CI. A test framework can be set using details given in this document.  Dimple Sharma  Software Products Intern |

**ArcGIS\_Collector\_Automation\_Project:**

**Mobile Test Automation using Appium**

The ArcGIS\_Collector automation projects is designed and developed(partially) to automate the Sanity Test run for Collector Daily build. It’s a Maven Project using TestNG and Jenkins CI.

**What is Appium?**

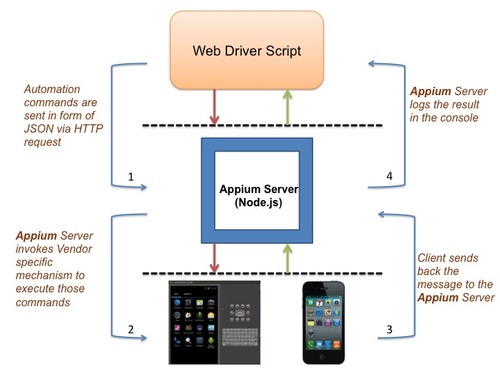
Appium is an open source test automation tool for mobile applications. It allows you to test all the three types of mobile applications: native, hybrid and mobile web. It also allows you to run the automated tests on actual devices, emulators and simulators.

ss

**Why Appium?**

* You shouldn’t have to recompile your app or modify it in any way in order to automate it.
* You shouldn’t be locked into a specific language or framework to write and run your tests.
* A mobile automation framework shouldn’t reinvent the wheel when it comes to automation APIs.
* A mobile automation framework should be open source, in spirit and practice as well as in name!

**Appium Architect:**



**Automation Environment Setup**

**Using Ruby:**

<https://devtopia.esri.com/apps/qa-common/wiki/Appium-Setup>

**Using Java:**

|  |  |
| --- | --- |
| **Software/Setups needed** | **Link to Software/Setups** |
| Java | <http://www.oracle.com/technetwork/java/javase/downloads/index.html> |
| Eclipse | <https://eclipse.org/downloads/> |
| Selenium Jars | <http://www.seleniumhq.org/download/> |
| Java clients jar | <https://mvnrepository.com/artifact/io.appium/java-client/2.1.0> |
| ssAppium Server | <http://appium.io/> |
| node JS | <https://nodejs.org/en/download/> |
| Android SDK | <https://developer.android.com/studio/index.html> |
| Device(phone/tablet) | Your Device |
| APK file | .apk File |

**Other important concepts used for automation framework set up:**

**TestNG:** TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. TestNG is designed to cover all categories of tests:  unit, functional, end-to-end, integration, etc.

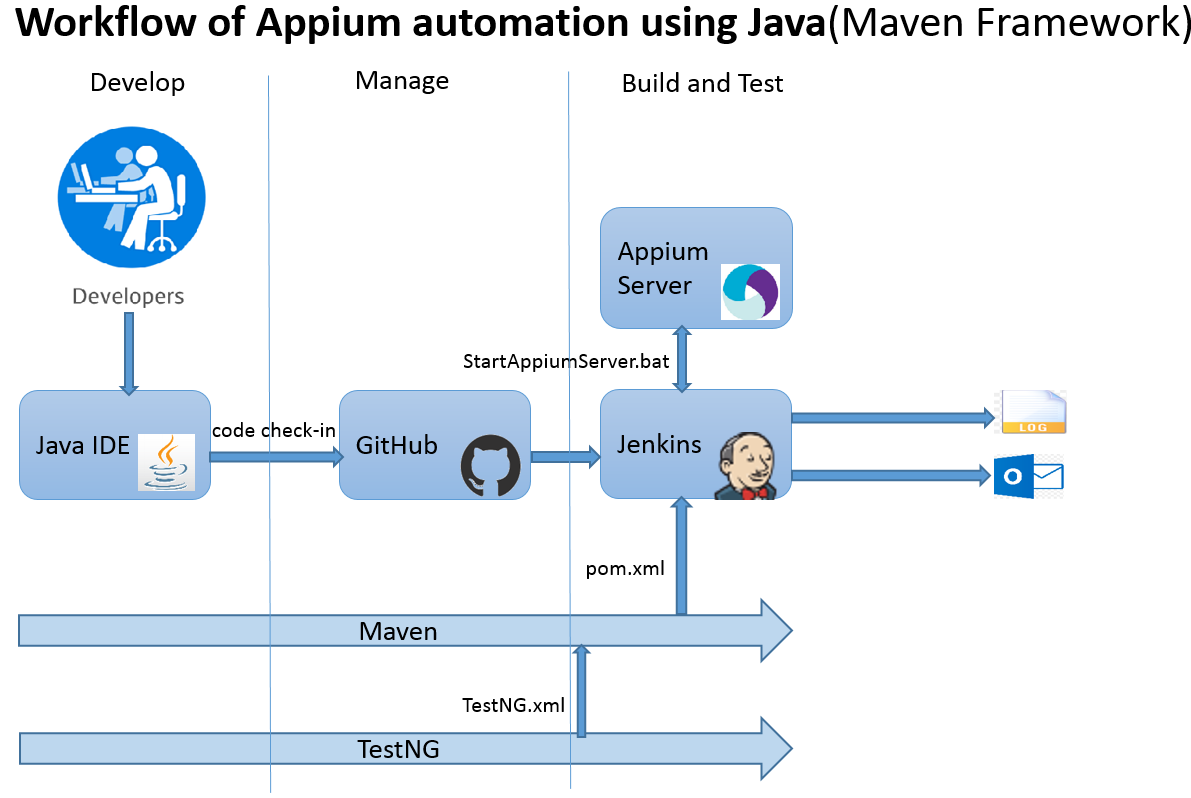
* For Running many test cases at one Trigger
* Execution cab be performed on class level as well as on Package level
* Regular expression

**Maven:** It isa build automation tool used primarily for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) projectsss

* Build Management tool
* For Creating Project Hierarchy or a Framework
* It takes care of required jar files

**Jenkins CI tool:** Jenkins is a Continuous Integration server. Basically Continuous Integration is the practice of running tests on a non-developer machine automatically every time someone pushes new code into the source repository. This has the tremendous advantage of always knowing if all tests work and getting fast feedback.

* For Build Scheduling
* Logs
* reporting



**Automation Source Code Path:**

[**\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\Appium\_Automation**](file:///\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\Appium_Automation)



**.bat script for starting the Appium Server:**

[**\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\Appium\_Automation\AppiumStartScript**](file:///\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\Appium_Automation\AppiumStartScript)

****

**Link to Collector Test Case Repository:** <https://devtopia.esri.com/apps/collector-test-catalog>

**Collector\_Cert\_Checklist\_V1.0 Template**

[**\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\TestingChecklist**](file:///\\devinfo\WRMResources\TestDocumentation\Mobile\Collector\TestingChecklist%20)

**Code description:**

|  |  |  |
| --- | --- | --- |
| **Class Name** | **Test case implemented** | **Comment** |
| startAppium.java | Starts the Appium server before every test case | BeforeTest annotation used |
|  |  |  |
| freshInstall.java | Install the current Build on the device |  |
| Start the application after installing is over |  |
| Check Learn More page |  |
| Capturing the snapshot of "Learn more Screen" |  |
| Check Try Collector |  |
| Opening the map under "All maps" |  |
| Installing/Launching the application |  |
|  |  |  |
| loginTest.java | Installing/Launching the application |  |
| Continue sign in to http://www.argis.com with incorrect credentials |  |
| Try login in with incorrect Credentials |  |
| Capture Screenshot of Login failed screen |  |
| Sign in with correct credentials |  |
|  |  |  |
| switch Account.java | installing/Launching the application |  |
| Sign in with correct credentials |  |
| Check/Capture the snapshot of About page |  |
| Switch the account |  |
| going back to previous screen | Android key events used |
|  |  |  |
| portalLogin.java | installing/Launching the application |  |
| Checking Portal (10.4/10.4.1/10.5) connectivity with IWA authentication |  |
| Capturing all logged in account in a snapshot |  |
|  |  |  |
| mobilefeature.java | installing/Launching the application |  |
| Sign in with correct credentials |  |
| swipe the group list to see if all groups loaded |  |

**Important piece of code:**

The Capabilities are defined and passed to Appium using below piece of code:

Appium server is triggered using a **StartAppiumServer.bat script**:

/\*

@BeforeTest

**public** **void** Appiumstart() **throws** IOException, InterruptedException

{

Runtime.*getRuntime*().exec("cmd /c start C:/Appium/StartAppiumServer.bat");

Thread.*sleep*(10000);

System.***out***.println("Appium started");

}

@Test

**public** **void** test() **throws** InterruptedException, IOException {

//below are the capabilities passed at the start of execution only

File AppDir = **new** File("src");

//place the .apk file in src folder

File app = **new** File(AppDir, "ArcGISCollector-release.apk");

DesiredCapabilities cap=**new** DesiredCapabilities();

// either provide ANDROID or iOS as mobile platform

cap.setCapability(MobileCapabilityType.***PLATFORM\_NAME***, MobilePlatform.***ANDROID***);

// Device name could be Emulator or the android device(in case of physical device)

cap.setCapability(MobileCapabilityType.***DEVICE\_NAME***, "Android Device");

cap.setCapability(MobileCapabilityType.***APP***, app.getAbsolutePath());

//This will avoid the timeout error for launch of start activity

cap.setCapability(MobileCapabilityType.***APP\_WAIT\_ACTIVITY***,"com.esri.arcgis.app.views.accounts.StartScreenActivity");

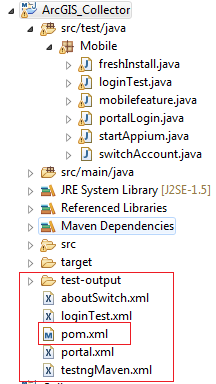
//cap.setCapability(MobileCapabilityType.APP\_WAIT\_ACTIVITY,"com.esri.arcgis.collector/com.esri.arcgis.app.framework.FirstActivity");

//This will avoid the timeout error if the page takes longer time to load

cap.setCapability(MobileCapabilityType.***NEW\_COMMAND\_TIMEOUT***, "1000");

\*/

**ssImportant configuration file:**

****

**Collector Sanity Test cases covered So far:**

1. Install the current Build on the device
2. Start the application after installing is over
3. Check Learn More page
4. Check Try Collector
5. Continue sign in to http://www.argis.com with incorrect credentials
6. Sign in with correct credentials check error message
7. go to right side Browse panel and select Switch Account--> check messages/prompts
8. Shutdown the app while signed in and check that credentials are remembered on next launch
9. Check the count of Maps in 'All Maps' tab
10. Open Maps from 'All Maps'
11. Check if all groups have loaded in browse panel list.
12. Check the map count in my maps & other groups to verify if we are pulling the correct maps count or not
13. Open the contents from my maps & other groups
14. Check About Page (Build#, Version, Links)
15. Login in to portals using IWA authentication