**Automation framework guide**

1. **Technology and tools used.**

Language: Java

Build tool: Maven

Test tool: TestNG

Automation tools: Appium, Selenium

Reporting tool: Allure (by Yandex) <http://allure.qatools.ru/>

IDE: IntelliJ IDEA‎ (Community)

Other: NPM (included in Node.JS)

1. **Environment setup**

* Download Java SE Development Kit, install it and setup JAVA\_HOME environment variable. Add JAVA\_HOME to PATH environment variable. (Reboot might be needed) Check if it is works correct by entering command line “java -vesrion”.
* Download Apache Maven, unpack it. Add the bin directory of the created directory apache-maven to the PATH environment variable. (Reboot might be needed) Check if it is works correct by entering command line – “mvn –v”.
* Download and install NPM (included in Node.JS). (Reboot might be needed) Check if it is works correct by entering command line – “npm –v”.

PATH Environment variable example:

echo %PATH%

C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\Sy

stem32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;**C:\Program Files\Java\jd**

**k1.8.0\_65\bin;**C:\Program Files\TortoiseSVN\bin;C:\Program Files\TortoiseHg\;**C:\Program Files\**

**nodejs\;C:\apache-maven-3.3.3\bin;** C:\Program Files\OpenV

PN\bin;C:\Program Files\Git\cmd;C:\MinGW\bin;C:\Program Files (x86)\Skype\Phone\

;C:\Python27;C:\Python27\scripts;**C:\Users\user\AppData\Roaming\npm**

* Download and install IntelliJ IDEA‎ (Community)

1. **Application under test**

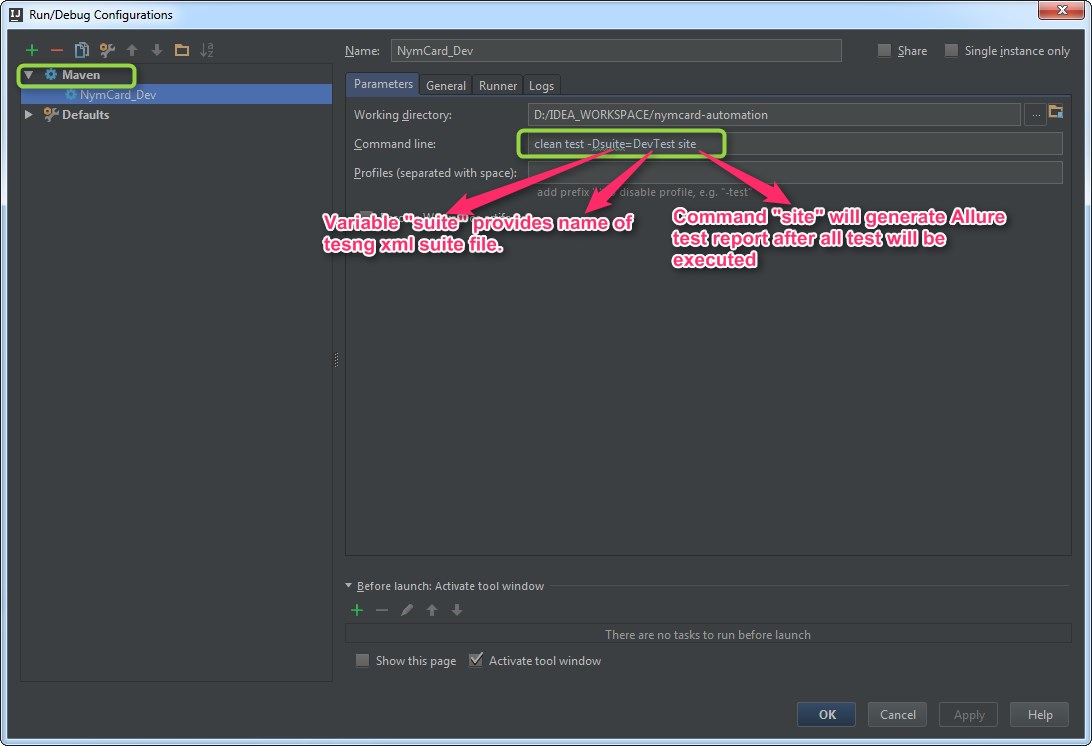
* Your Apple ID should be added to Adept team
* Real iPad device should be also add to devices of Adept team
* Get Apple iOS development certificate for Adept team and provisions profile
* Install them on MAC machine where Appium will be used to run tests
* Get sources of Adept iPad application
* Build application from sources on real iPad device (which will be used for auto-testing)
* After build done successful, go to Xcode DerivedData and get .app file of build done on previous step
* Drop this .app file on opened iTunes window
* Then, drug and drop this file from iTunes back to your desktop – in result you will have .ipa file
* This .ipa file will be our application under test (like the one that was used here: /Volumes/IMAC\_Automation/adept/ADEPT\_RealDevice.ipa)

1. **Appium installation for iOS**

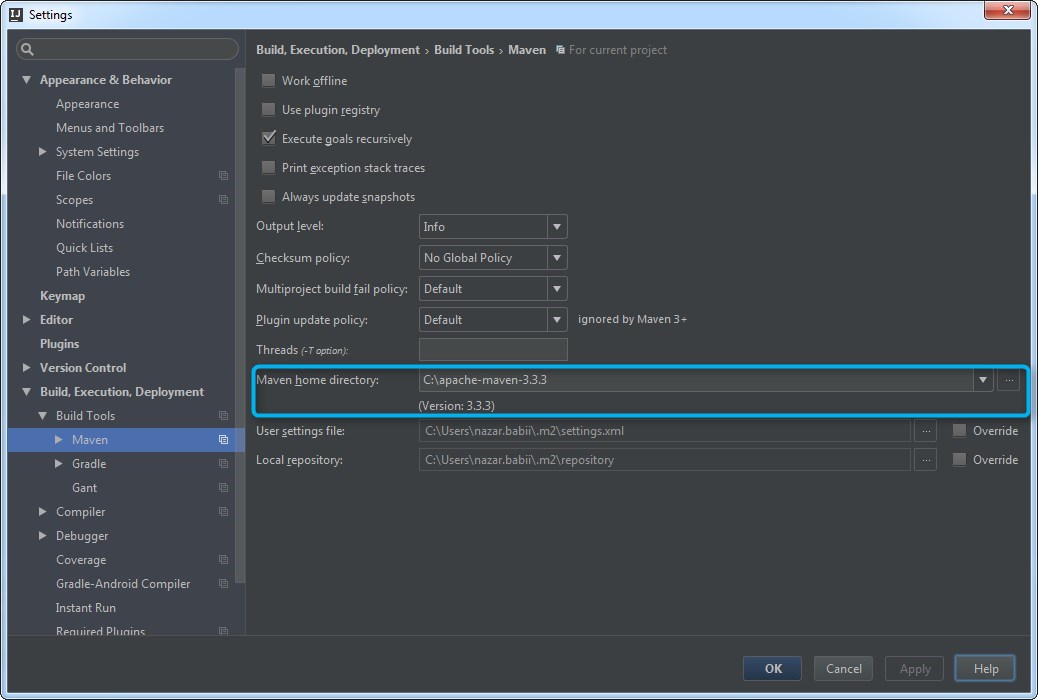
* Clone Git Appium repo, and checkout latest released revision. <https://github.com/appium/appium/releases>
* Setup Appium according to <https://github.com/appium/appium/blob/master/docs/en/contributing-to-appium/appium-from-source.md> (for iOS)
* Set up appium-xcuitest-driver according to <https://github.com/appium/appium-xcuitest-driver>
* Set up configuration for running tests on real device according to <https://github.com/appium/appium-xcuitest-driver/blob/master/docs/real-device-config.md>. Application under test should be build using same Apple ID which was used in this step.

1. **IDE setup**

* Open project using IntelliJ IDEA‎.
* To run Maven project under IntelliJ IDEA, you need to create run configuration as following



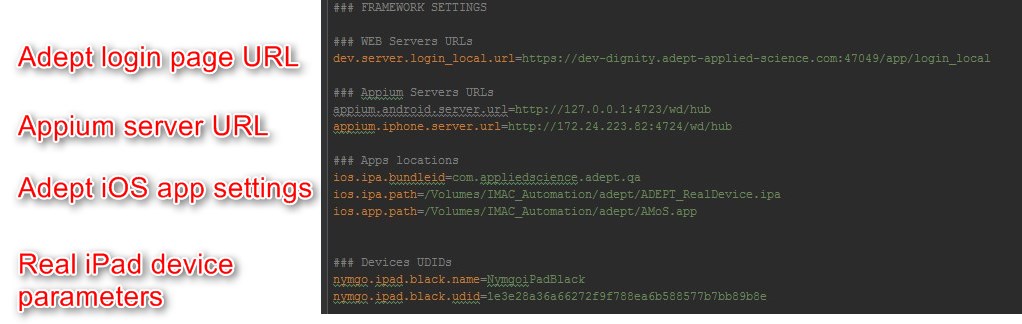
* Setup Maven path in IntelliJ IDEA settings



1. **Framework settings**

All framework settings should be provided using **settings.properties**.

Current version looks following:



1. **First test run**

* Suppose you have Appium is on remote MAC machine and framework is on your machine. So **settings.properties** should have: appium.iphone.server.url=http://<MAC machine IP>:PORT/wd/hub.
* Suppose you just need to run some single test. So, you need to mark test, which you want to run, with group “demo”
* Open terminal on MAC machine with Appium installed. Start Appium with command like “appium --address <MAC IP> --port <PORT> --default-capabilities '{"nativeInstrumentsLib":true}'”. You will see response like:



* iPad application under test should be also placed on MAC machine. For example here: /Volumes/IMAC\_Automation/adept/ADEPT\_RealDevice.ipa
* Connect iPad to the MAC machine with Appium
* Update device settings in **settings.properties** according to connected device
* In project in IntelliJ IDEA execute run configuration that was created before (in a section 5), it will run tests marked with group “demo”. In result test should be run on device.
* After all tests done, you can check detailed report in \target\site\allure-maven-plugin.html. HTML file should be opened using **Firefox** browser.
  + **If all tests were successful**, Allure report will be generated automatically, and could be found in \target\site\allure-maven-plugin.html.
  + **If some tests were failed**, Allure report could be generated manually after all tests finished. Just open command line, navigate to project folder and execute command “mvn site”. Report will be generated to \target\site\allure-maven-plugin.html.

1. **Web and Mobile testing setup (implementation not finished)**

Framework could handle testing of iPad application and also testing of Web portal. In a BaseTest class before each test we can handle which part should be tested (iPad or web). For driver setup could be used **MobileDriverManager** or **WebDriverManager**.

So each test in XML file (like DevTest.xml) could have parameters which will be parsed in BaseTest-> BeforeTest and appropriate setup will be done. Parameters which needed, you can find as parameters in methods: webDriverManager.getInstance and mobileDriverManager.initialiseDriver.