YOUR QUALITY PARTNER FOR SOFTWARE SOLUTIONS

Robot framework

Web Automation - (Selenium for Desktop & Appium for Mobile)

Tho Pham

Version: 0.6

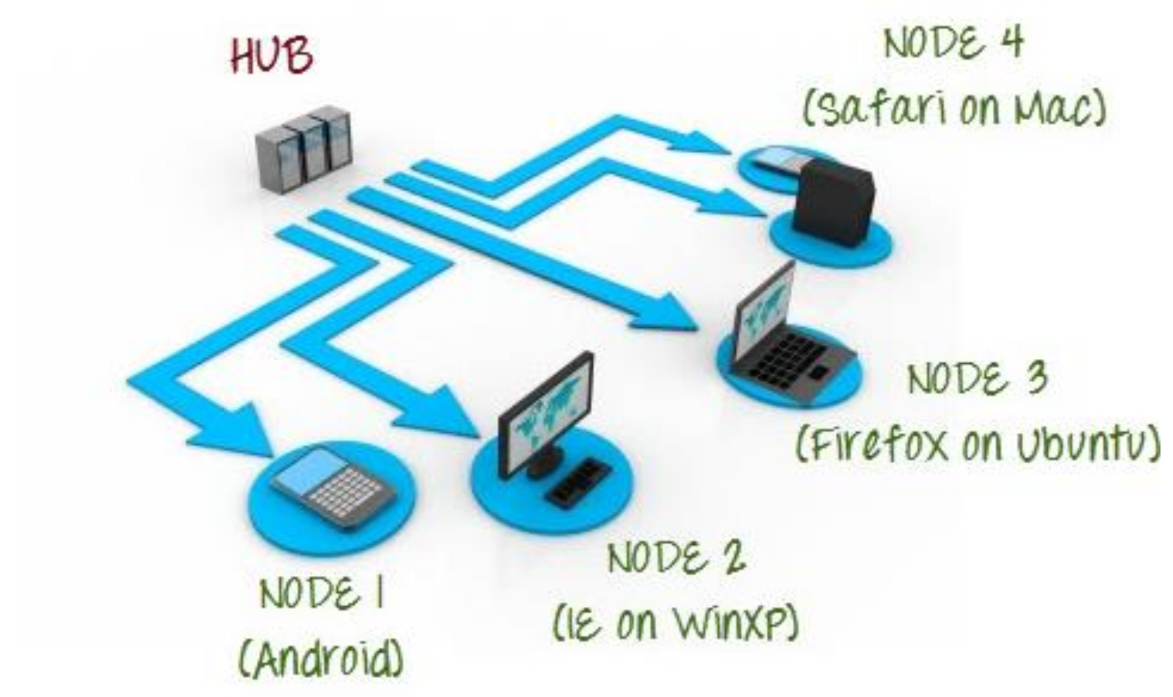
Agenda

Duration: 6 hours with 2 parts

- **Prerequisites:**
  - Have knowledge on Robot Framework https://box.tma.com.vn/index.php/s/KxPUddTu5e6Pqf4 and https://box.tma.com.vn/index.php/s/J3Rto4XOiuoYJwo
  - Have experiences on Selenium https://box.tma.com.vn/index.php/s/PDJAY74PwfYANww
  - Download ZIP https://github.com/phamtantho/robotframework
    - On Windows: Put on D:/ and unzip
    - On Mac: Put on Desktop/ and unzip; go inside selenium_grid > run cmd: chmod 755 *
  - Have Macbook, iOS and Android devices
- **Part 1 - Desktop:**
  - Introduce Selenium Grid
  - Test web on windows (Chrome, Firefox, Edge)
  - Test web on MacOS (Chrome, Firefox, Safari)
  - Page Object Model (POM)
- **Part 2 - Mobile:**
  - Test web on Android (Chrome)
  - Test web on iOS (Safari)
  - Bonus: Headless browsers

Selenium Grid
Introduction

- Selenium-Grid allows you run your tests on different machines against different browsers and operating systems in parallel
- More details: *https://www.seleniumhq.org/docs/07_selenium_grid.jsp*



Selenium Grid

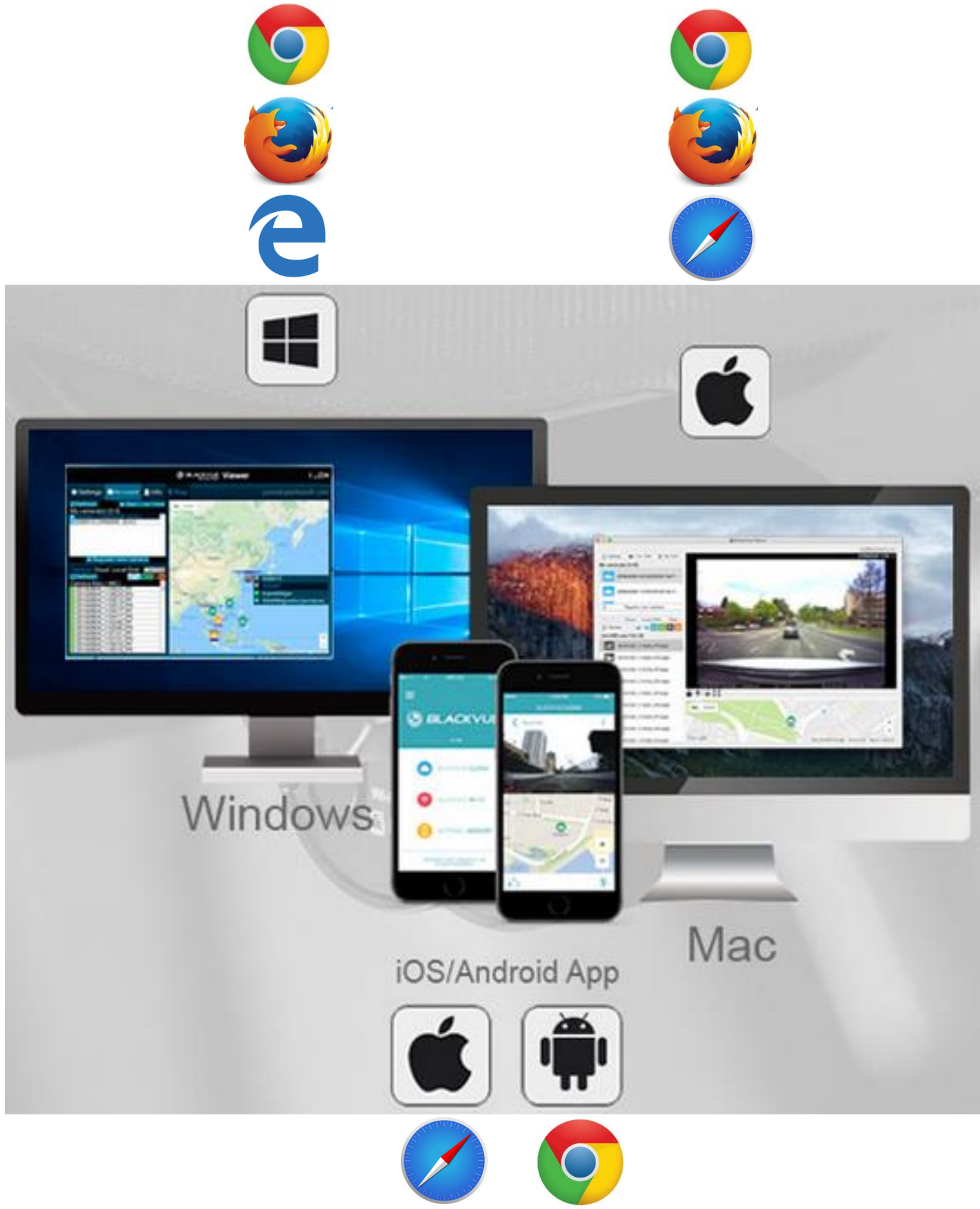
Architecture

- The Hub
  - The hub is the central point where you load your tests into.
  - There should only be one hub in a grid.
  - The hub is launched only on a single machine, i.g a computer whose O.S is Windows 10
  - The machine containing the hub is where the tests will be run, but you will see the browser being automated on the node.

Selenium Grid
Architecture

The Nodes

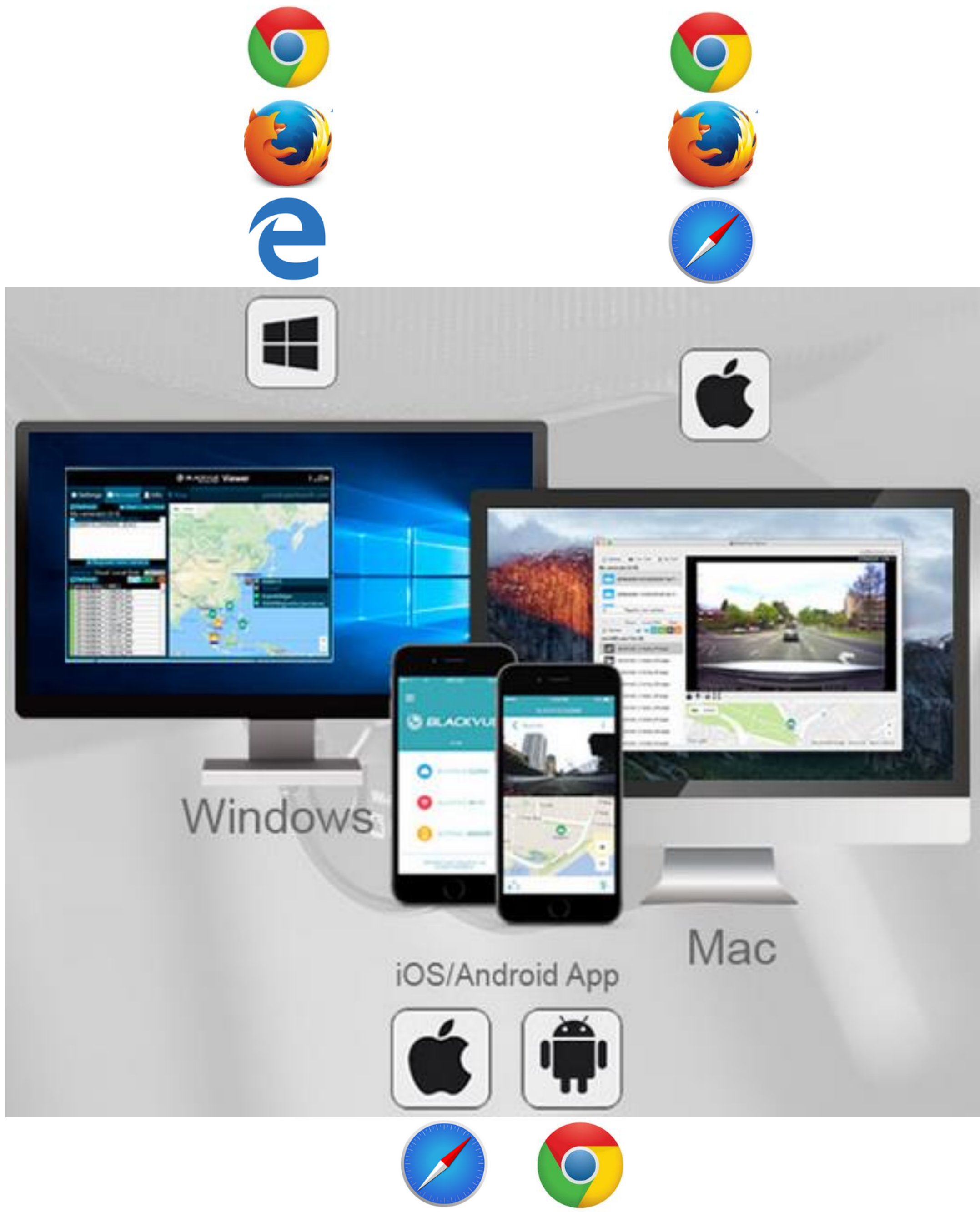
- Nodes are the Selenium instances that will execute the tests that you loaded on the hub.
- There can be one or more nodes in a grid.
- Nodes can be launched on multiple machines with different platforms and browsers.
- The machines running the nodes need not be the same platform as that of the hub.



Selenium Grid
Architecture

The Nodes

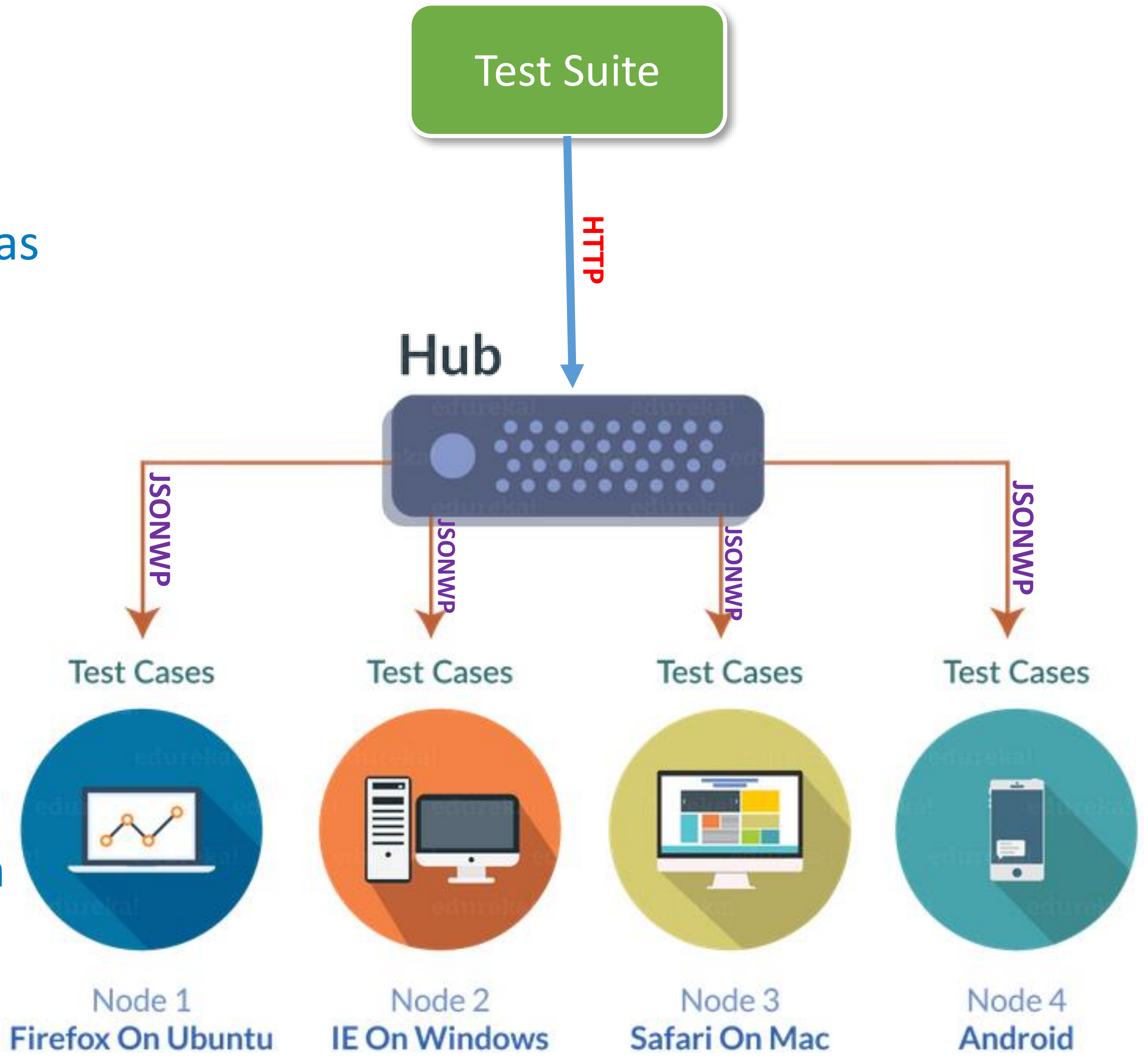
- Nodes are the Selenium instances that will execute the tests that you loaded on the hub.
- There can be one or more nodes in a grid.
- Nodes can be launched on multiple machines with different platforms and browsers.
- The machines running the nodes need not be the same platform as that of the hub.



Selenium Grid

How it works

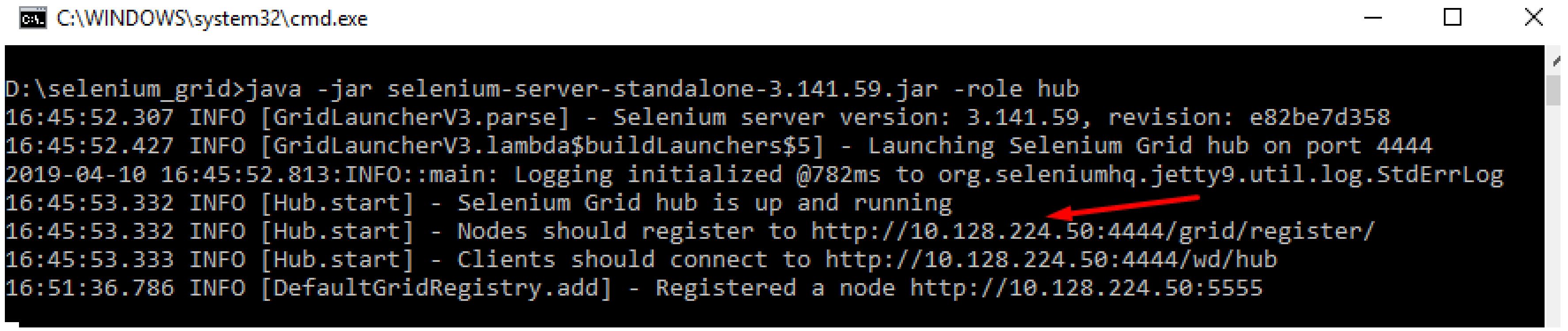
1. The hub receives a test to be executed along with information on which browser and platform
2. The hub 'knows' the configuration of each node that has been 'registered' to it
3. The hub selects an available node that has the requested browser-platform combination
4. Once a node has been selected, Selenium commands initiated by the test are sent to the hub, which passes them to the node assigned to that test
5. The node runs the browser, and executes the Selenium commands within that browser against the application under test.



Selenium Grid

Start hub

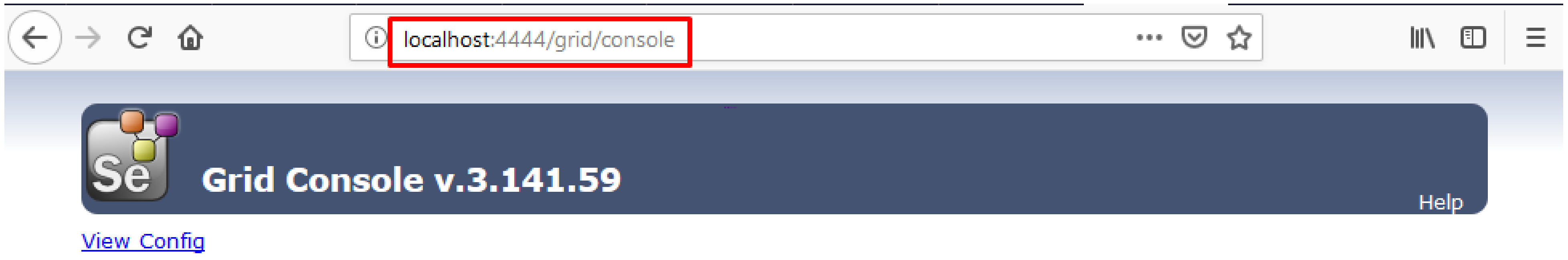
- Go to folder selenium_grid
  - On windows: double-click **start_hub_on_windows.bat**
  - On mac: open terminal, go to selenium_grid and run **./start_hub_on_mac.sh**
- Take note the hub's IP. In this case 10.128.224.50



C:\WINDOWS\system32\cmd.exe

```
D:\selenium_grid>java -jar selenium-server-standalone-3.141.59.jar -role hub
16:45:52.307 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
16:45:52.427 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2019-04-10 16:45:52.813:INFO::main: Logging initialized @782ms to org.seleniumhq.jetty9.util.log.StdErrLog
16:45:53.332 INFO [Hub.start] - Selenium Grid hub is up and running
16:45:53.332 INFO [Hub.start] - Nodes should register to http://10.128.224.50:4444/grid/register/
16:45:53.333 INFO [Hub.start] - Clients should connect to http://10.128.224.50:4444/wd/hub
16:51:36.786 INFO [DefaultGridRegistry.add] - Registered a node http://10.128.224.50:5555
```

- Verify hub status: Open page *<localhost or hub's IP>:4444/grid/console*



localhost:4444/grid/console

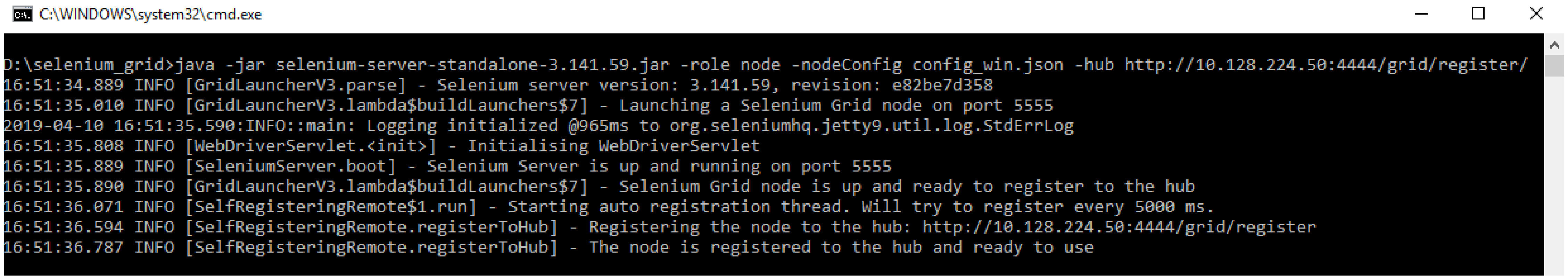
Grid Console v.3.141.59

Help

View Config

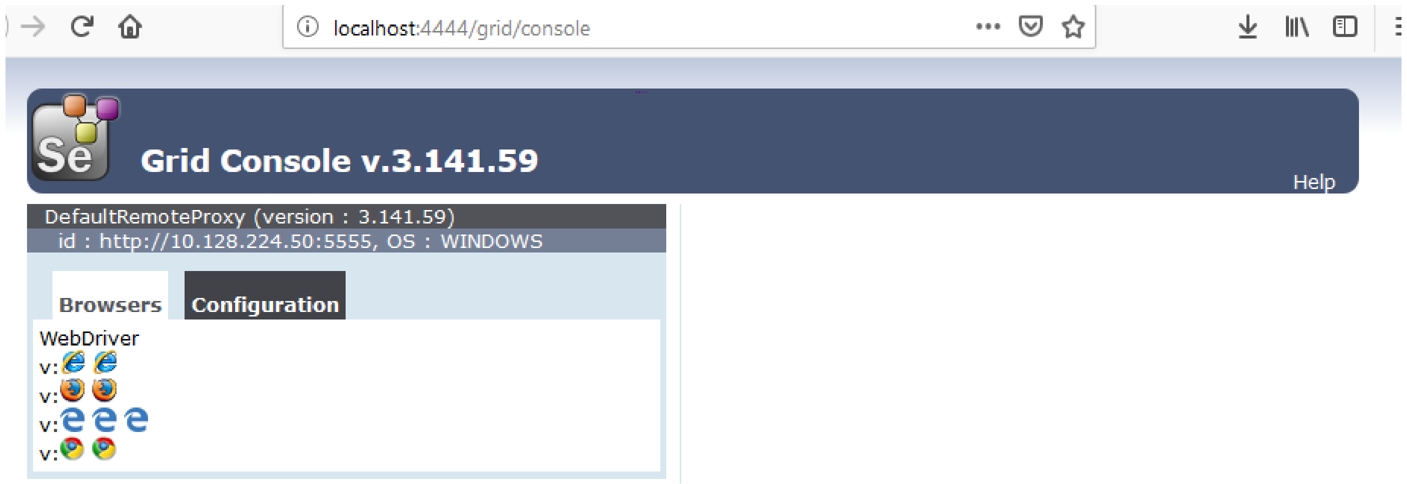
Test web on desktop
Start node on windows

- Go to selenium_grid
  - Update hub’s IP in file config_win.json
    - Line: "hub": "http://<hub’s IP>:4444",
  - Double click **start_node_on_windows.bat**



C:\WINDOWS\system32\cmd.exe

```
D:\selenium_grid>java -jar selenium-server-standalone-3.141.59.jar -role node -nodeConfig config_win.json -hub http://10.128.224.50:4444/grid/register/
16:51:34.889 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
16:51:35.010 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5555
2019-04-10 16:51:35.590:INFO::main: Logging initialized @965ms to org.seleniumhq.jetty9.util.log.StdErrLog
16:51:35.808 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
16:51:35.889 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5555
16:51:35.890 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
16:51:36.071 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
16:51:36.594 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://10.128.224.50:4444/grid/register
16:51:36.787 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
```



localhost:4444/grid/console

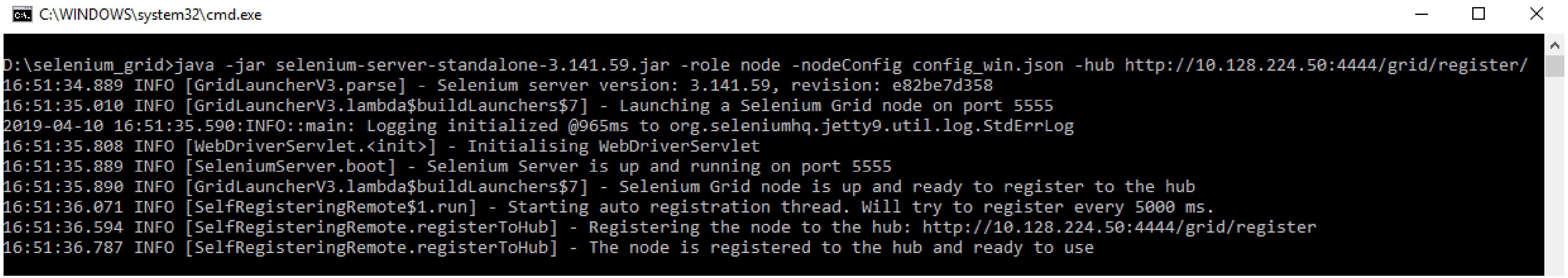
Grid Console v.3.141.59

DefaultRemoteProxy (version : 3.141.59)
id : http://10.128.224.50:5555, OS : WINDOWS

Test web on desktop

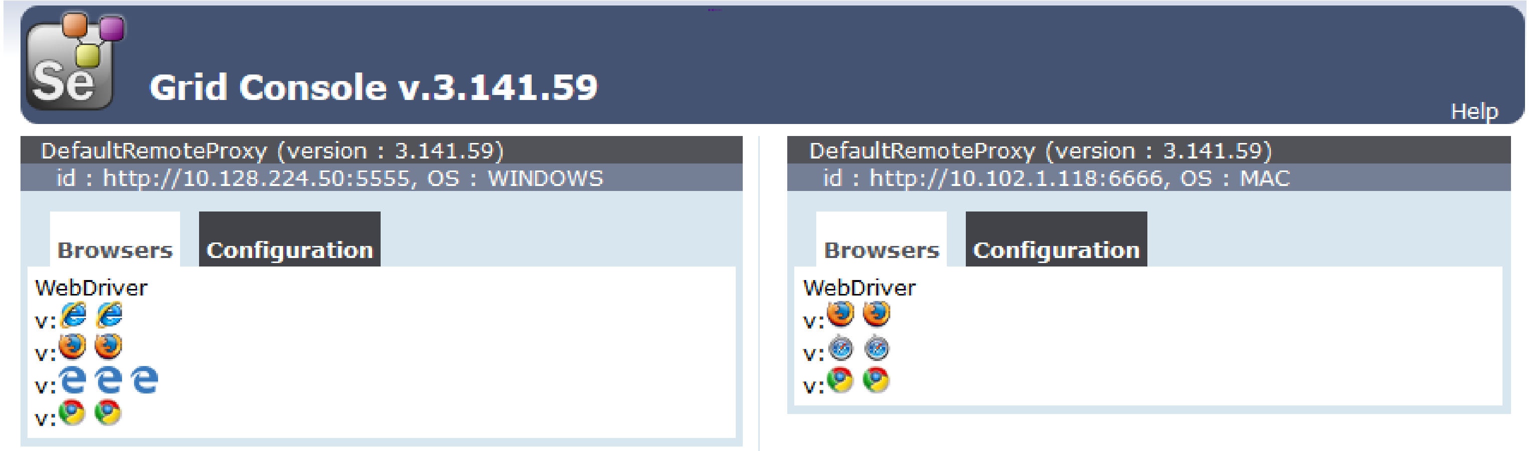
Start node on mac

- Go to selenium_grid
  - Update hub's IP in file start_node_on_mac.sh
    - Line: "hub": "http://<hub's IP>:4444",
  - Run cmd **./start_node_on_mac.sh**



C:\WINDOWS\system32\cmd.exe

```
D:\selenium_grid>java -jar selenium-server-standalone-3.141.59.jar -role node -nodeConfig config_win.json -hub http://10.128.224.50:4444/grid/register/
16:51:34.889 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
16:51:35.010 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5555
2019-04-10 16:51:35.590:INFO::main: Logging initialized @965ms to org.seleniumhq.jetty9.util.log.StdErrLog
16:51:35.808 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
16:51:35.889 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5555
16:51:35.890 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
16:51:36.071 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
16:51:36.594 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://10.128.224.50:4444/grid/register
16:51:36.787 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
```


Se
Grid Console v.3.141.59
Help
DefaultRemoteProxy (version : 3.141.59)
id : http://10.128.224.50:5555, OS : WINDOWS
Browsers
Configuration
WebDriver
v:
v:
v:
v:
DefaultRemoteProxy (version : 3.141.59)
id : http://10.102.1.118:6666, OS : MAC
Browsers
Configuration
WebDriver
v:
v:
v:

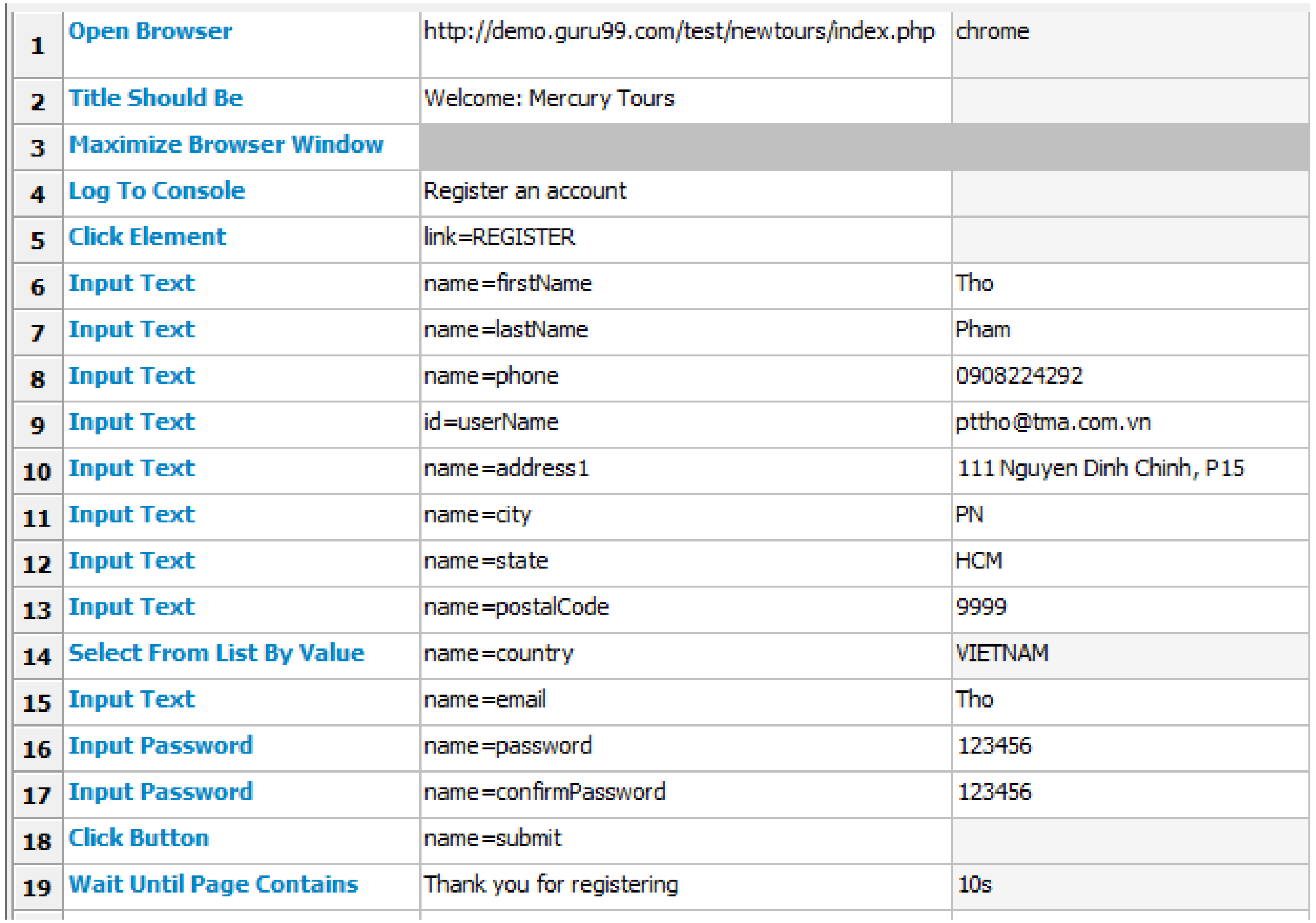
Sample test

Sample test 1

- Go to Robot_web_testing > test_browsers
- Take a look on this sample test
- Execute and observe the results
- Add more nodes from nearby classmates
  - PC1: Hub
  - PC2: node1
  - PC3: node2
  - …
- Execute and observe the results

Sample test
Disadvantages

- Duplicated code
  - Duplicated functionality
  - Duplicated locators
- Consequences
  - Fragile project
  - Less maintainable
  - Hard to read and follow

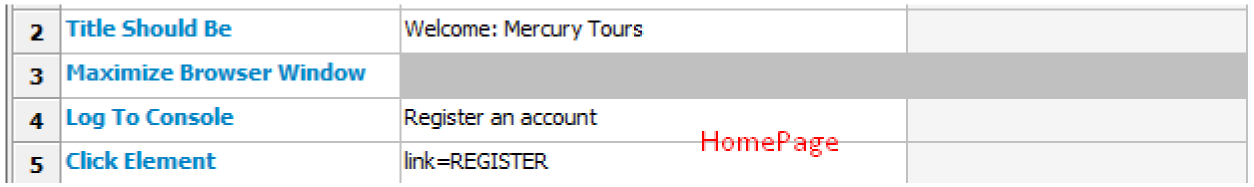


1	Open Browser	http://demo.guru99.com/test/newtours/index.php	chrome
2	Title Should Be	Welcome: Mercury Tours	
3	Maximize Browser Window		
4	Log To Console	Register an account	
5	Click Element	link=REGISTER	
6	Input Text	name=firstName	Tho
7	Input Text	name=lastName	Pham
8	Input Text	name=phone	0908224292
9	Input Text	id=userName	pttho@tma.com.vn
10	Input Text	name=address1	111 Nguyen Dinh Chinh, P15
11	Input Text	name=city	PN
12	Input Text	name=state	HCM
13	Input Text	name=postalCode	9999
14	Select From List By Value	name=country	VIETNAM
15	Input Text	name=email	Tho
16	Input Password	name=password	123456
17	Input Password	name=confirmPassword	123456
18	Click Button	name=submit	
19	Wait Until Page Contains	Thank you for registering	10s

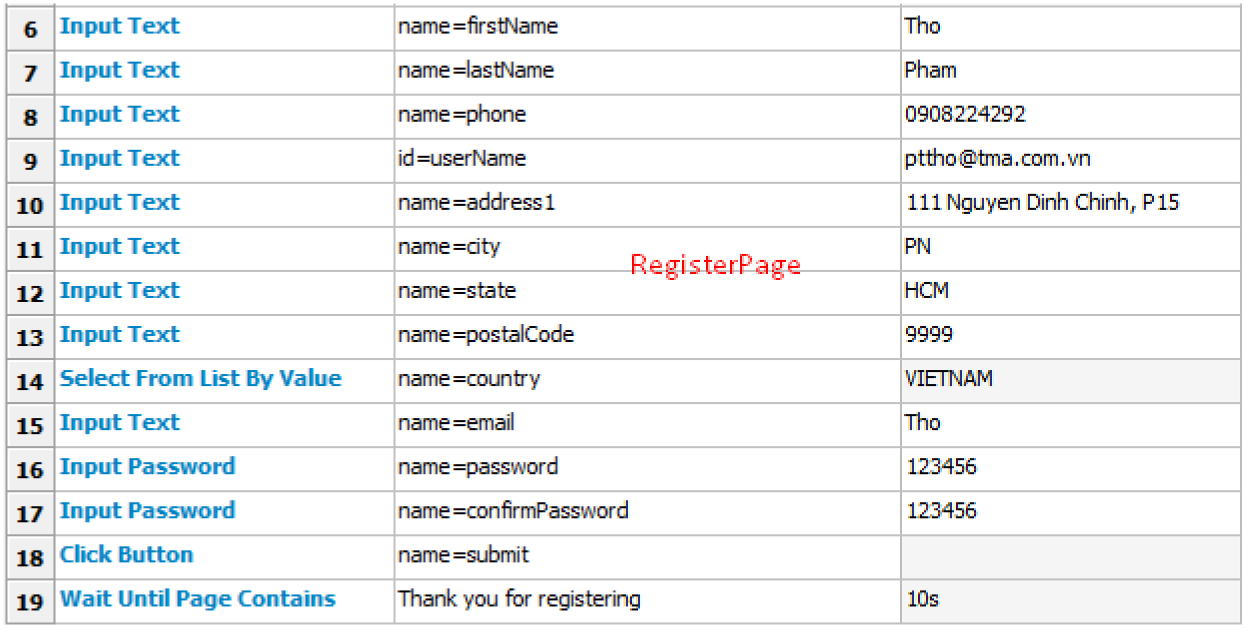
Page Object Model (POM)

What is POM and its advantages

- POM is a design pattern to create **Object Repository** for web UI elements
- Each web page in the application has corresponding page class that contains:
  - WebElements
  - Methods



2	Title Should Be	Welcome: Mercury Tours	
3	Maximize Browser Window		
4	Log To Console	Register an account	
5	Click Element	link=REGISTER	


RegisterPage

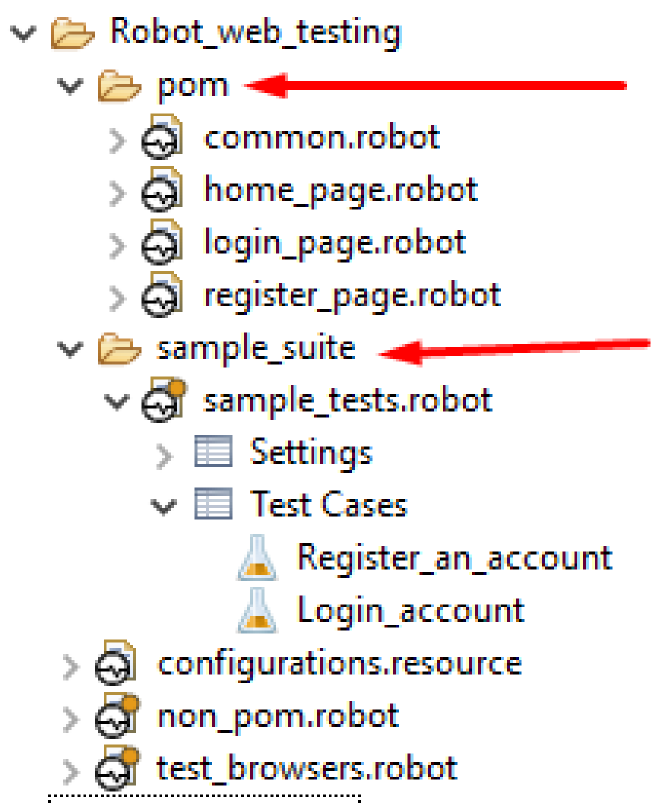
6	Input Text	name=firstName	Tho
7	Input Text	name=lastName	Pham
8	Input Text	name=phone	0908224292
9	Input Text	id=userName	pttho@tma.com.vn
10	Input Text	name=address1	111 Nguyen Dinh Chinh, P15
11	Input Text	name=city	PN
12	Input Text	name=state	HCM
13	Input Text	name=postalCode	9999
14	Select From List By Value	name=country	VIETNAM
15	Input Text	name=email	Tho
16	Input Password	name=password	123456
17	Input Password	name=confirmPassword	123456
18	Click Button	name=submit	
19	Wait Until Page Contains	Thank you for registering	10s

Reference: https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html

Page Object Model (POM)

Sample test 2

- Go to Robot_web_tesing > pom and sample_suite
- Take a look on test scripts
- Execute on remote nodes and observe results

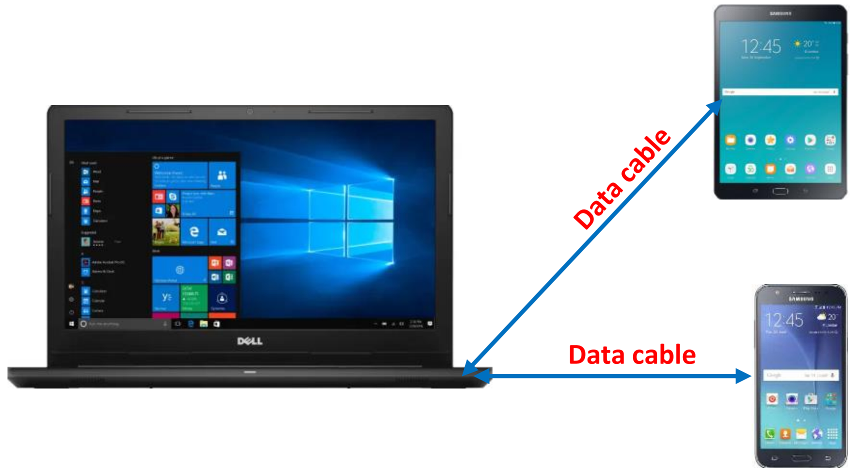


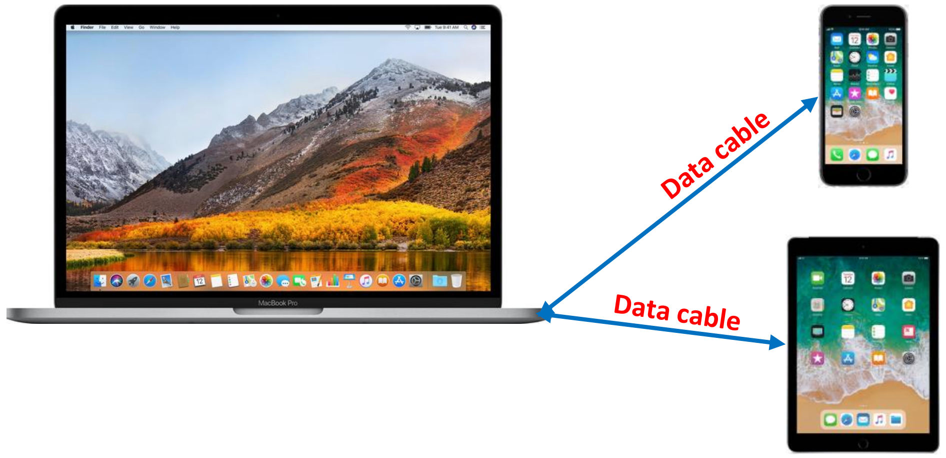
```
Robot_web_testing
  pom
    common.robot
    home_page.robot
    login_page.robot
    register_page.robot
  sample_suite
    sample_tests.robot
      Settings
      Test Cases
        Register_an_account
        Login_account
  configurations.resource
  non_pom.robot
  test_browsers.robot
```

Test web on mobile devices

Prerequisites

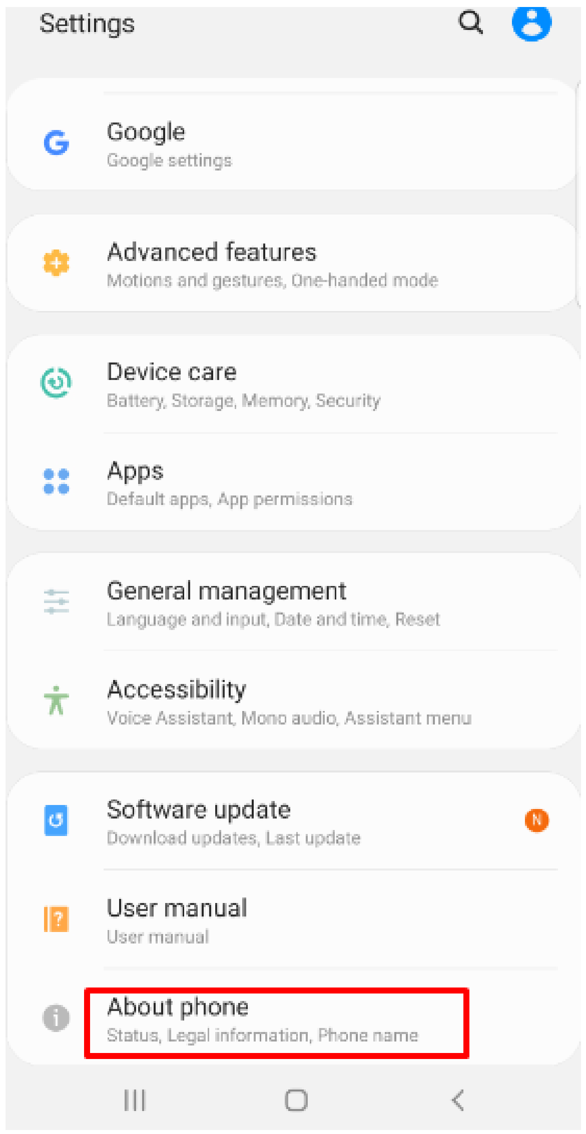
- Android
  - Windows machine
  - Android device (mobile or tablet) + Data cable
  - ADB (Android Debug Bridge)
- iOS
  - MacOS machine with **OS 10.13.6**
  - iOS device (ipad or iphone) **OS >=10** + Data cable
  - **Xcode 10.1** & Command Line Tools
  - Apple account





Test web on Android devices

Setup for Android – Enable Developer options



Settings

Google
Google settings

Advanced features
Motions and gestures, One-handed mode

Device care
Battery, Storage, Memory, Security

Apps
Default apps, App permissions

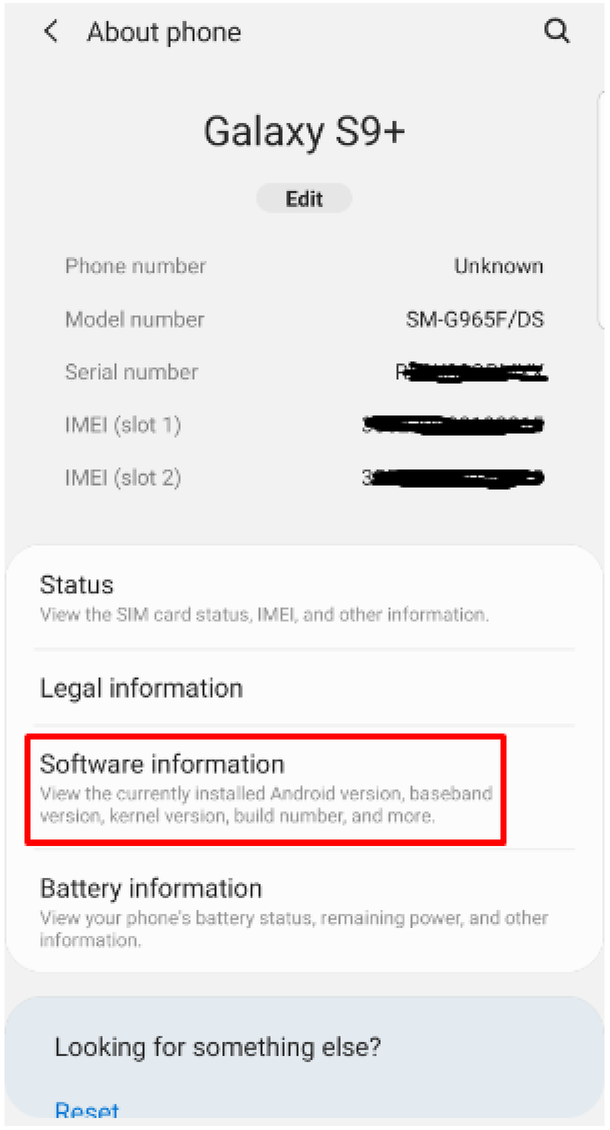
General management
Language and input, Date and time, Reset

Accessibility
Voice Assistant, Mono audio, Assistant menu

Software update
Download updates, Last update

User manual
User manual

About phone
Status, Legal information, Phone name



About phone

Galaxy S9+

Edit

Phone number	Unknown
Model number	SM-G965F/DS
Serial number	[redacted]
IMEI (slot 1)	[redacted]
IMEI (slot 2)	[redacted]

Status
View the SIM card status, IMEI, and other information.

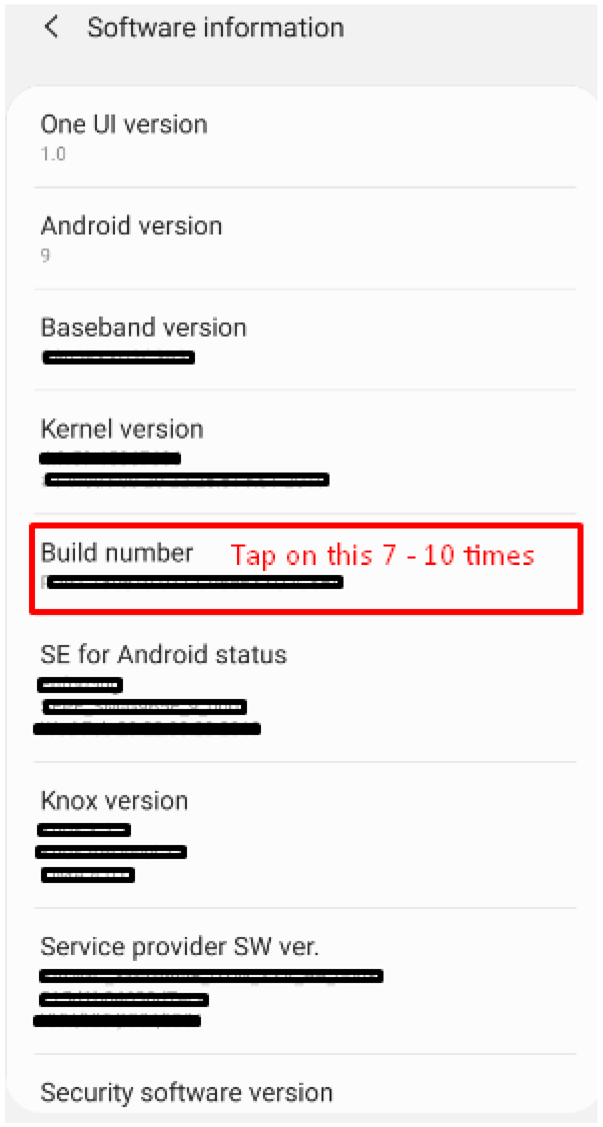
Legal information

Software information
View the currently installed Android version, baseband version, kernel version, build number, and more.

Battery information
View your phone's battery status, remaining power, and other information.

Looking for something else?

Reset



Software information

One UI version
1.0

Android version
9

Baseband version
[redacted]

Kernel version
[redacted]

Build number Tap on this 7 - 10 times
[redacted]

SE for Android status
[redacted]

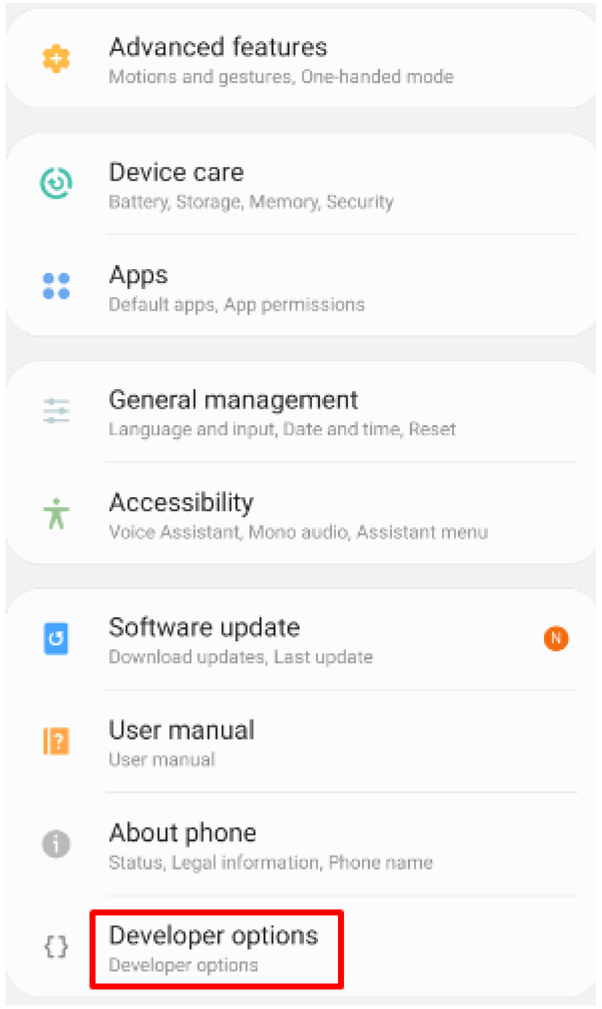
Knox version
[redacted]

Service provider SW ver.
[redacted]

Security software version

Test web on Android devices

Setup for Android– Enable USB Debugging



Advanced features
Motions and gestures, One-handed mode

Device care
Battery, Storage, Memory, Security

Apps
Default apps, App permissions

General management
Language and input, Date and time, Reset

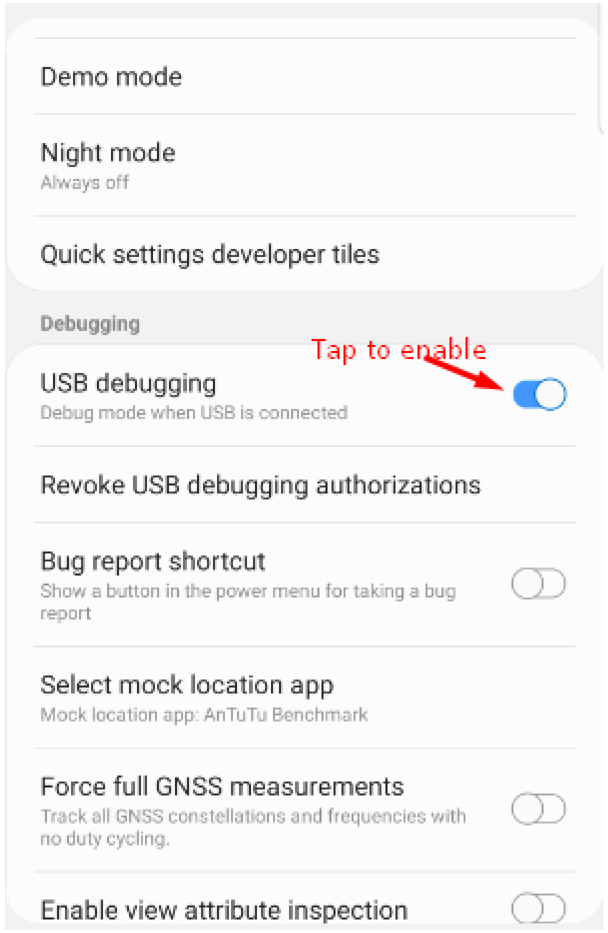
Accessibility
Voice Assistant, Mono audio, Assistant menu

Software update
Download updates, Last update

User manual
User manual

About phone
Status, Legal information, Phone name

Developer options
Developer options



Demo mode

Night mode
Always off

Quick settings developer tiles

Debugging

USB debugging
Debug mode when USB is connected

Tap to enable

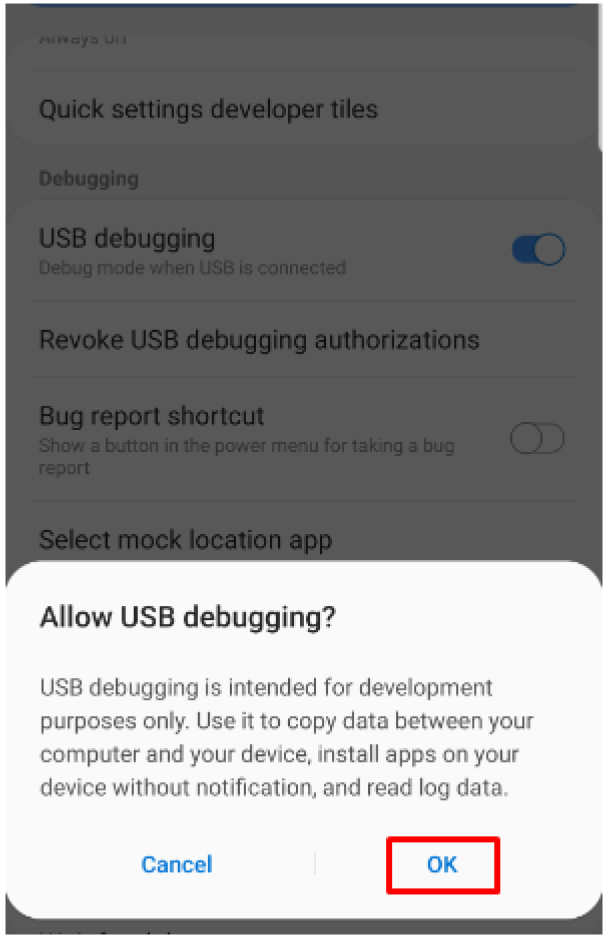
Revoke USB debugging authorizations

Bug report shortcut
Show a button in the power menu for taking a bug report

Select mock location app
Mock location app: AnTuTu Benchmark

Force full GNSS measurements
Track all GNSS constellations and frequencies with no duty cycling.

Enable view attribute inspection



Quick settings developer tiles

Debugging

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

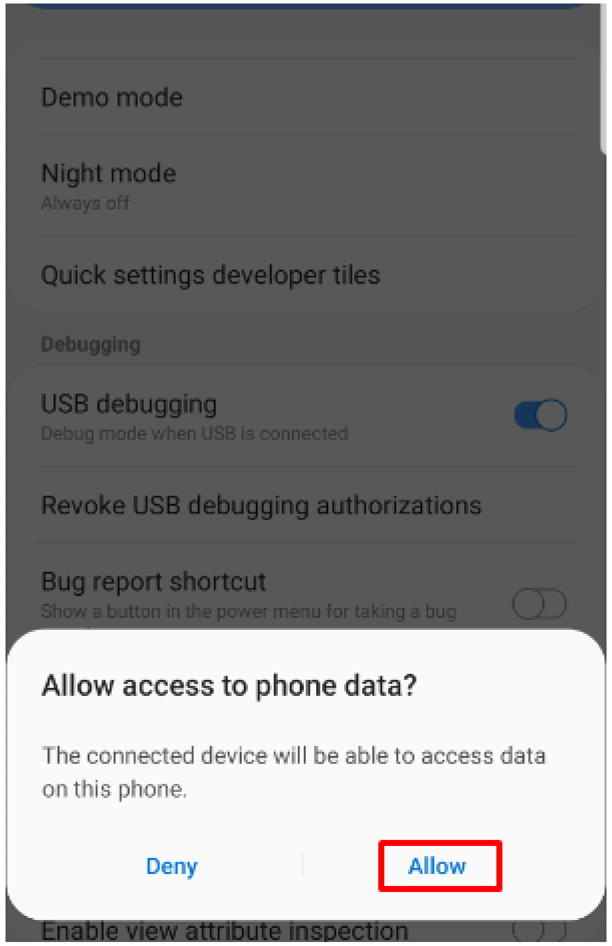
Bug report shortcut
Show a button in the power menu for taking a bug report

Select mock location app

Allow USB debugging?

USB debugging is intended for development purposes only. Use it to copy data between your computer and your device, install apps on your device without notification, and read log data.

Cancel OK



Demo mode

Night mode
Always off

Quick settings developer tiles

Debugging

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

Bug report shortcut
Show a button in the power menu for taking a bug

Allow access to phone data?

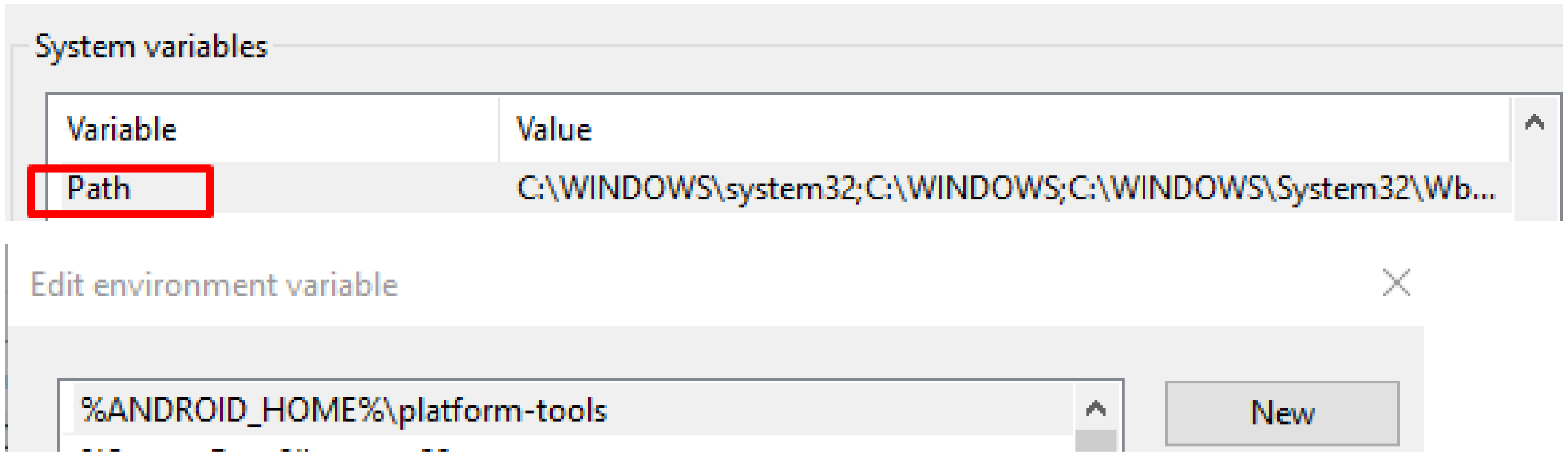
The connected device will be able to access data on this phone.

Deny Allow

Enable view attribute inspection

Test web on Android devices
Setup for Android

- On Windows machine:
  - Go to robotframework-master > SDK.zip > go inside and get SDK folder path
  - Add advanced system variable **ANDROID_HOME** = SDK path (i.g D:\ robotframework-master\SDK)
  - Add **%ANDROID_HOME%\platform-tools** to **Path**
  - Test setup: Open cmd, type command adb devices → Should show connected device. Otherwise, double check the configurations above.


System variables

Variable	Value
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wb...

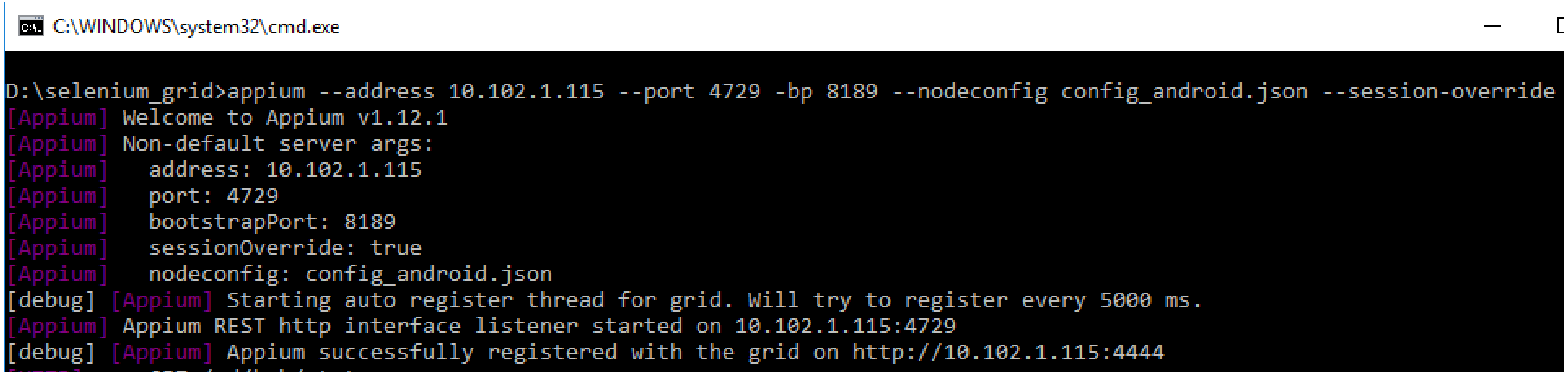
Edit environment variable

%ANDROID_HOME%\platform-tools

Test web on Android devices

Start node on Android

- Go to selenium_grid
  - **Start hub if not yet started**
  - Update hub's Port and IP in file config_android.json on these lines:
    - "url": "http://<IP of windows machine where android connects to>/wd/hub" i.g "url": http://10.102.1.115:4729/wd/hub
    - "host": "<IP of windows machine where android connects to>" i.g "host": "10.102.1.115",
    - "hubPort": "<hub's Port>" i.g "hubPort": "4444"
    - "hubHost": "<hub's IP>" i.g "hubHost": "10.102.1.115"
  - Update IP of windows where android connects to after **--address** in **start_node_android_on_windows.bat**
    - i.g --address 10.102.1.115
  - Double click **start_node_android_on_windows.bat**



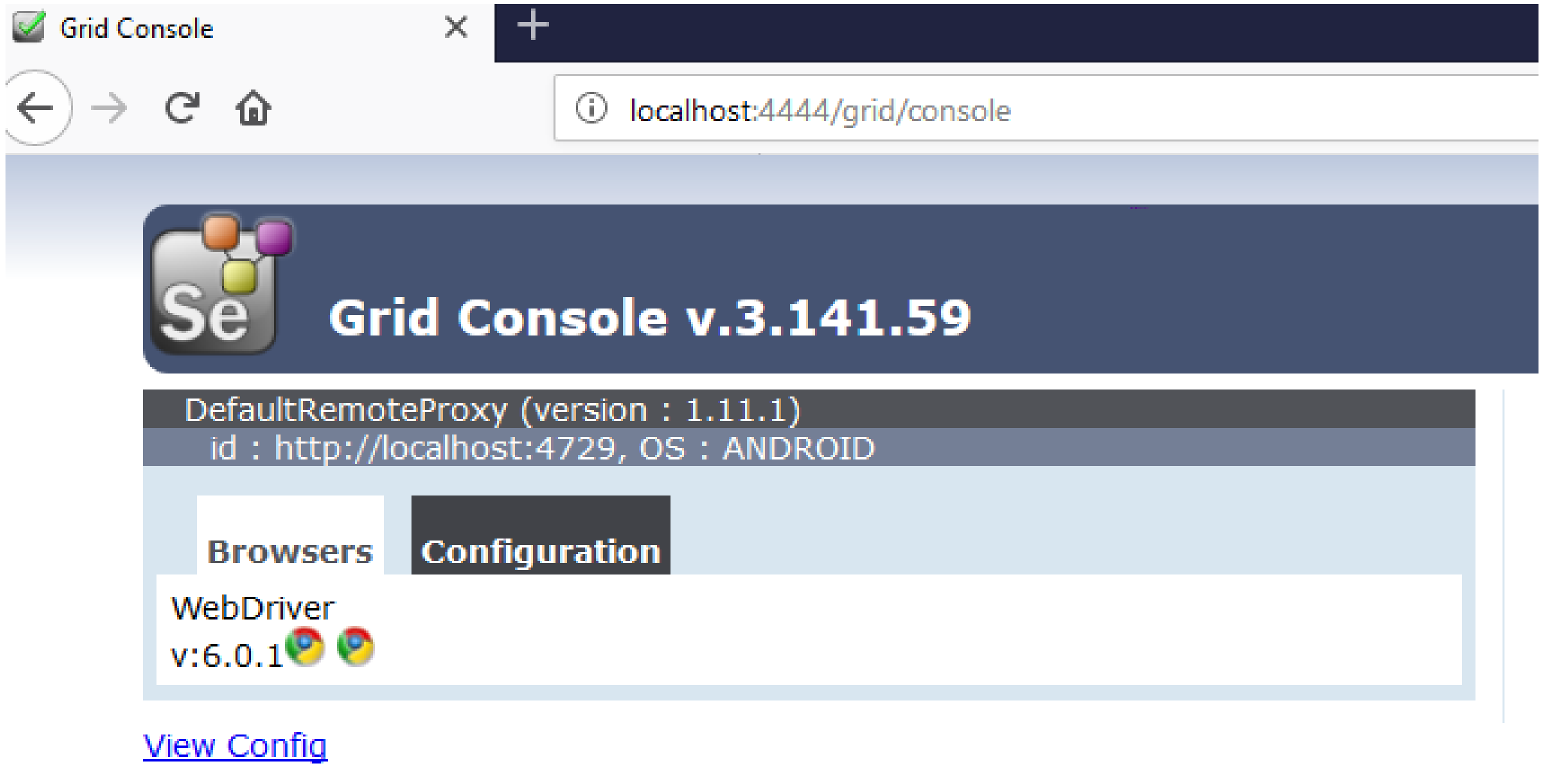
C:\WINDOWS\system32\cmd.exe

```
D:\selenium_grid>appium --address 10.102.1.115 --port 4729 -bp 8189 --nodeconfig config_android.json --session-override
[Appium] Welcome to Appium v1.12.1
[Appium] Non-default server args:
[Appium]   address: 10.102.1.115
[Appium]   port: 4729
[Appium]   bootstrapPort: 8189
[Appium]   sessionOverride: true
[Appium]   nodeconfig: config_android.json
[debug] [Appium] Starting auto register thread for grid. Will try to register every 5000 ms.
[Appium] Appium REST http interface listener started on 10.102.1.115:4729
[debug] [Appium] Appium successfully registered with the grid on http://10.102.1.115:4444
```

Test web on Android devices

Start node on Android

- On browser, go to http://<localhost or hub's IP>:4444/grid/console



Grid Console v.3.141.59

DefaultRemoteProxy (version : 1.11.1)
id : http://localhost:4729, OS : ANDROID

WebDriver v:6.0.1

Sample test

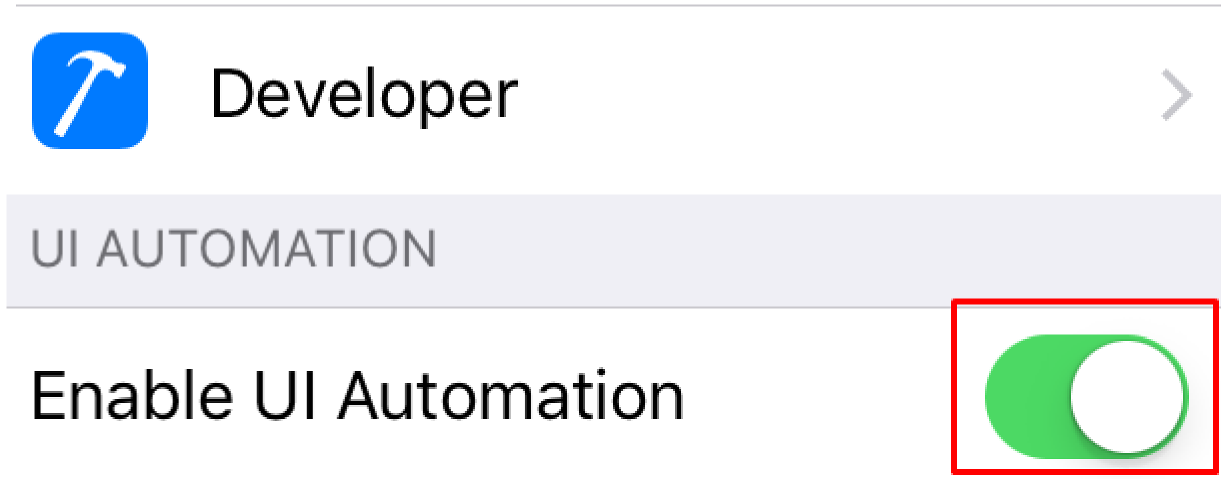
Sample test 3

- Go to Robot_web_testing > test_browsers
- Take a look on test cases:
  - Test_chrome_on_android
  - Register_an_account_on_android
  - Login_account_on_android
- Execute and observe the results

Test web on iOS devices

Setup for iOS

- Enable Developer Mode on iOS
  - Connect iOS device to Macbook. Click Trust if popup appears.
  - Open Settings>Developer>Enable UI Automation



Note: If you do not see Developer in Settings, open Xcode on Mac and connect your iOS device to Macbook and check for Developer again in Settings.

Test web on iOS devices

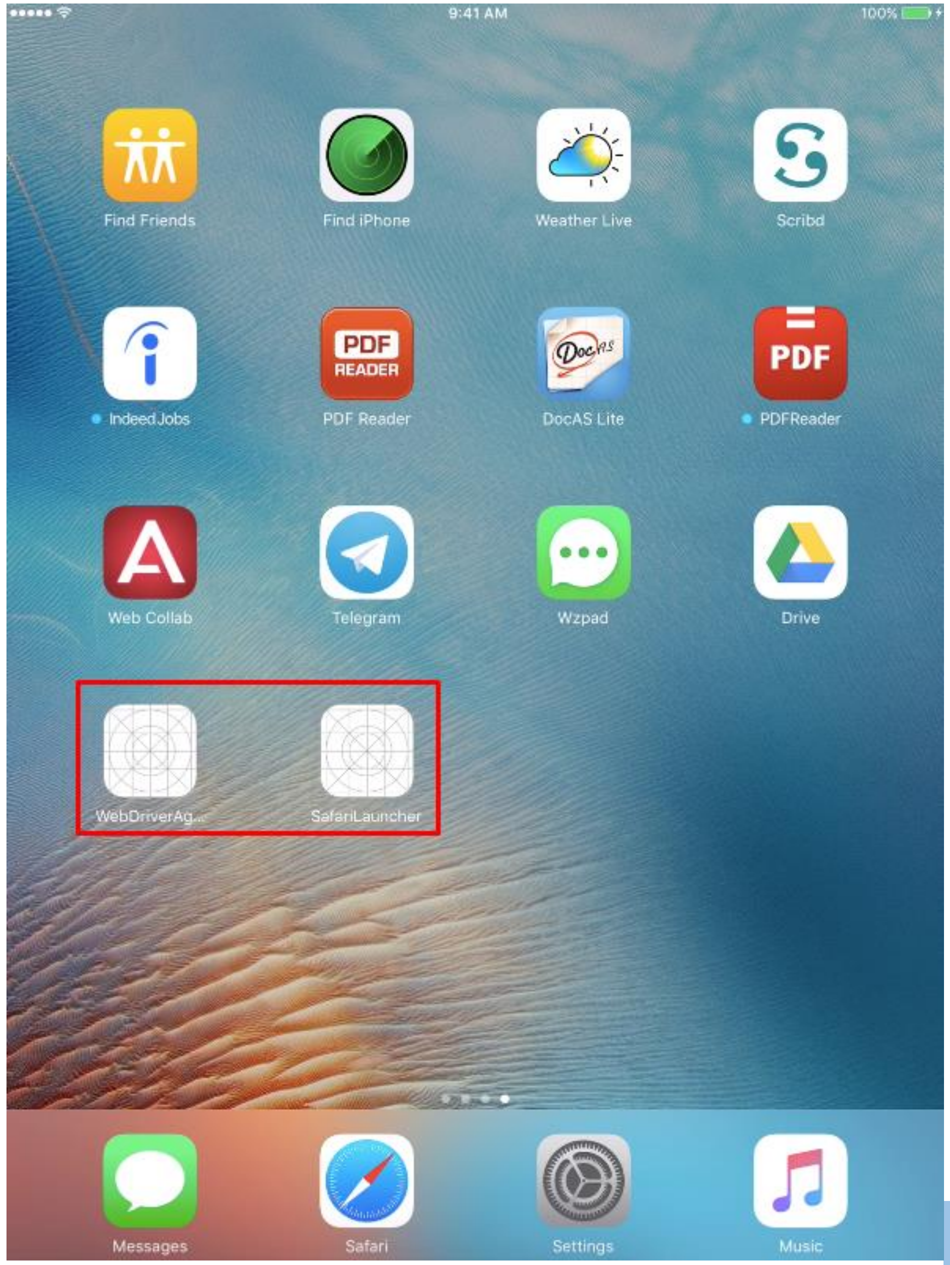
Setup for iOS

- On MacOS machine:
  - Upgrade to Xcode 10.1 ← MacOS 10.13.6
  - Install brew: **/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"**
  - Install npm: **brew install node**
  - Install carthage: **brew install carthage**
  - Install Appium cmd tool: **sudo npm install –g appium --unsafe-perm=true**
  - XCUITest Driver Real Device Setup: http://appium.io/docs/en/drivers/ios-xcuitest-real-devices/
  - SafariLauncher Setup: http://appium.io/docs/en/drivers/ios-uiautomation-safari-launcher/index.html

Test web on iOS devices

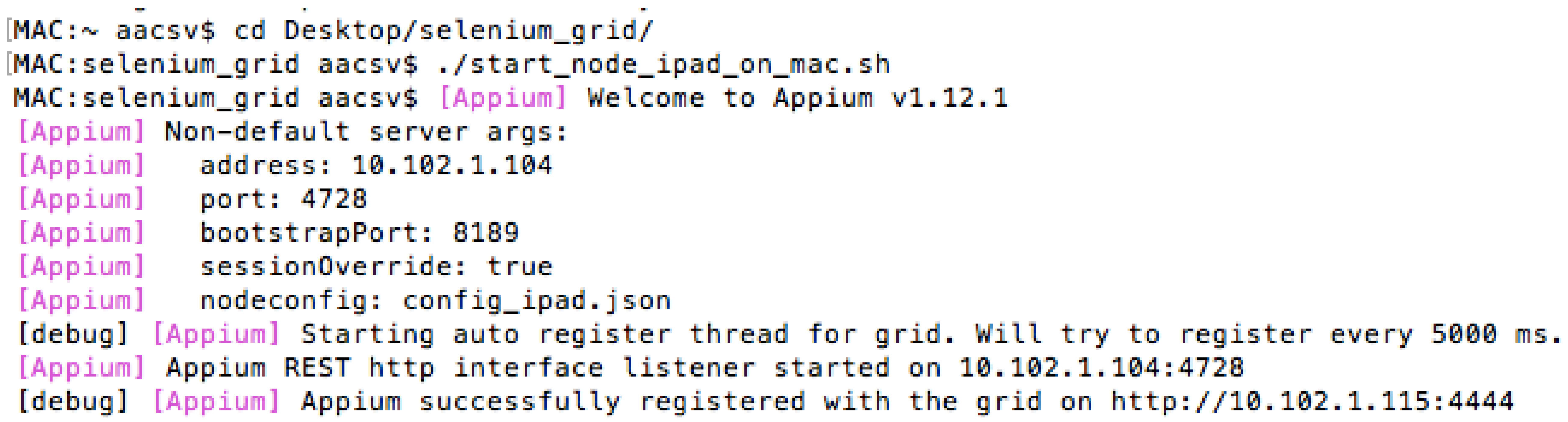
Setup for iOS

- On iOS device, ensure these 2 applications are installed:
  - WebDriverAgent
  - SafariLauncher



Test web on iOS devices
Start node on iOS

- Terminal: ios_webkit_debug_proxy –c <udid>:27753 –d
  - I.g: ios_webkit_debug_proxy –c 3e1946f3c1ff3591fa72e20b84a4a70c38df0f09:27753 –d
- Terminal > go to selenium_grid
  - Update hub's Port and IP in file config_ipad.json on these lines:
    - "url": http://<IP of Mac machine where iPad connects to>:4729/wd/hub i.g "url": http://10.102.1.104:4729/wd/hub
    - "host": "<IP of Mac machine where iPad connects to> " i.g "host": "10.102.1.104"
    - "hubPort": "<hub's Port>" i.g "hubPort": "4444"
    - "hubHost": "<hub's IP>" i.g "hubHost": "10.102.1.115"
  - Update IP of Macbook where iOS connects to after **--address** in **start_node_ipad_on_mac.sh**
  - Execute ./**start_node_ipad_on_mac.sh**

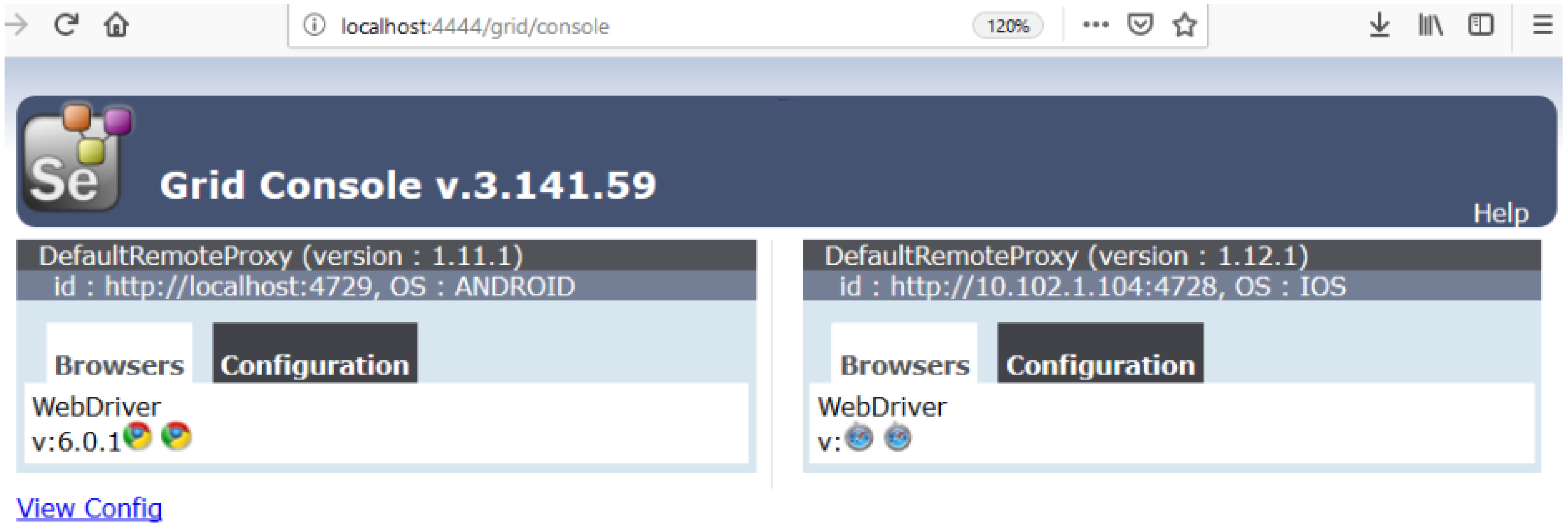


```
MAC:~ aacsv$ cd Desktop/selenium_grid/
MAC:selenium_grid aacsv$ ./start_node_ipad_on_mac.sh
MAC:selenium_grid aacsv$ [Appium] Welcome to Appium v1.12.1
 [Appium] Non-default server args:
 [Appium]   address: 10.102.1.104
 [Appium]   port: 4728
 [Appium]   bootstrapPort: 8189
 [Appium]   sessionOverride: true
 [Appium]   nodeconfig: config_ipad.json
 [debug] [Appium] Starting auto register thread for grid. Will try to register every 5000 ms.
 [Appium] Appium REST http interface listener started on 10.102.1.104:4728
 [debug] [Appium] Appium successfully registered with the grid on http://10.102.1.115:4444
```

Remote driver

Selenium Grid for Desktop – Verify Grid

- Open a browser and go to http://<localhost or hub's IP>:4444/grid/console



localhost:4444/grid/console

Grid Console v.3.141.59

Help

DefaultRemoteProxy (version : 1.11.1)
id : http://localhost:4729, OS : ANDROID

Browsers | Configuration

WebDriver
v:6.0.1

DefaultRemoteProxy (version : 1.12.1)
id : http://10.102.1.104:4728, OS : IOS

Browsers | Configuration

WebDriver
v:

View Config

Sample test

Sample test 4

- Go to Robot_web_testing > test_browsers
- Take a look on test cases:
  - Test_safari_on_ios
  - Register_an_account_on_ios
  - Login_account_on_ios
- Execute and observe the results

Headless Browser

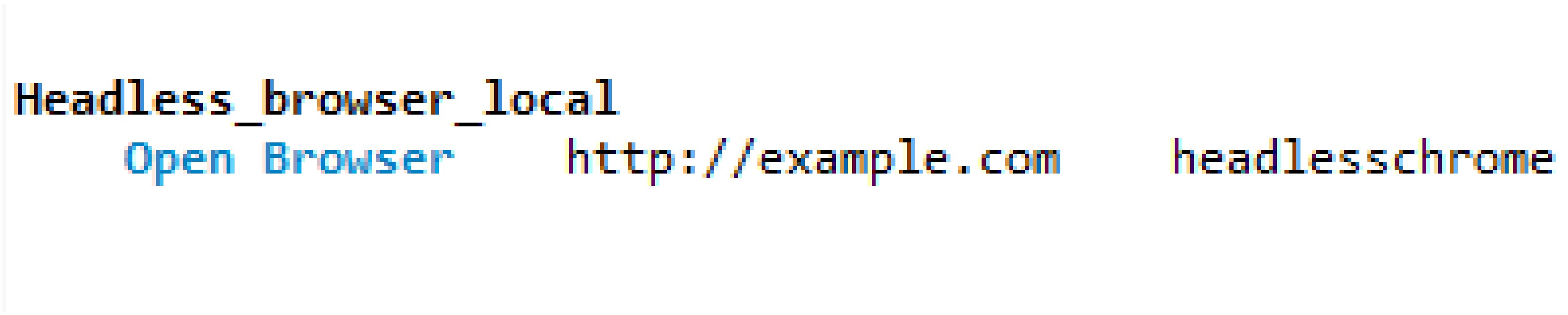
Introduction

- A headless browser is a web-browser without a graphical user interface. This program will behave just like a browser but will not show any GUI
- Robot supports 4 kinds of headless browser
  - Chrome
  - Firefox
  - HtmlUnit
  - PhantomJS
- More details:

  http://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Open%20Browser

Headless Browser
How to use

- In keyword Open Browser, use following browser names to test headless browser:
  - Chrome: headlesschrome
  - Firefox: headlessfirefox
  - HtmlUnit: htmlunit
  - PhantomJS: phantomjs
- Example:



```
Headless_browser_local
    Open Browser    http://example.com    headlesschrome
```

Robot Framework

Q&A

THANK YOU !

Tel:	+84 8 3997-8000	North America number:	+ 1 802-735-1392
Mobile:	+84 908-676-212	Australia number:	+ 61 414-734-277
Fax:	+84 8 3990-3303	Japan number:	+81 3-6432-4994
Email:	sales@tmasolutions.com	Website:	www.tmasolutions.com