Browser http://192.168.56.1:4444/grid/console#

// \*\*\*\* HOW TO RUN PARALLEL GRID \*\*\*\*  
//1 Create json config files for each device  
//2 Open 3 cli  
//3 java -jar selenium-server-standalone-3.0.0-beta4.jar -role hub  
//NO Appium server must be started for each device and then we execute the commands to register it with the selenium server standalone  
//4 open new cli and navigate to the appium folder for each device new cli, arguments in quotaion marks  
//5 node.exe node\_modules\appium\bin\appium.js --nodeconfig "C:\Users\ceko\IdeaProjects\demo\src\main\resources\appium\_node\_S4.json" -p 4723 -U d9e1470c  
//6 node.exe node\_modules\appium\bin\appium.js --nodeconfig "C:\Users\ceko\IdeaProjects\demo\src\main\resources\appium\_node\_S6.json" -p 4724 -U 0715f7c98061163a  
*hosts*.put("d9e1470c", new URL("http://127.0.0.1:4723/wd/hub"));  
*hosts*.put("0715f7c98061163a", new URL("http://127.0.0.1:4724/wd/hub"));

[Configuring Appium Node Servers with Selenium Grid](http://blog.zymr.com/configuring-appium-node-servers-with-selenium-grid)

Posted by [Pavan O.](http://blog.zymr.com/author/pavan-o) on Nov 17, 2015 8:30:00 AM

* [inShare](javascript:void(0);)107



Mobile applications are becoming widely available on the market at the speed of light while development and testing of particular applications on multiple platforms and devices becomes a challenge. There can be multiple platforms (Android, iOS) and multiple devices which a Mobile App can support. However it involves a lot of resources and time for development to cover all of the tests across these devices.

Mobile Test Automation is a solution to save time and resources. In this use case we will cover the steps needed to configure Appium servers with Selenium Grid to run Mobile Test Automation. The steps explained in this use case will be similar when working with iOS devices. We will also cover how multiple mobile real/virtual devices connected/hosted on the same machine or distributed across different machines can be driven with the Appium framework and Selenium Grid. This use case will cover the steps to configure Appium with Selenium Grid for multiple mobile devices.

We will see how two android virtual devices (Samsung S4 and S5, or you can use any other brand of devices listed in Genymotion) are configured with the Appium server and Selenium Grid to be part of the test automation infrastructure.



GOING OVER THE DIAGRAM

*Appium Client Library* – The Client library used to script the test automation scripts to communicate with the Appium server.

*Selenium Grid* – Central Hub whose job would mostly be to redirect requests to the correct Appium server and in turn to the correct Device.

*Appium Servers* – Appium servers are nothing but the Node.js server hosted which can understand the Appium framework requests coming in the form of Webdriver REST APIs. These Appium Servers communicate with the respective Android devices configured with them in the form of JSON Wired Protocol.

*Genymotion AVDs*– The Genymotion is software which can host multiple android devices which is as good as attaching the real device to the particular machine. Note that default AVDs which come with Android SDK or any other AVDs providing software will also work for the scenario explained in the use case.

**[See Also:**[Appium: iOS UI Software Test Automation](http://blog.zymr.com/appium-ios-ui-automation)**]**

ASSUMPTIONS TO FIRST MAKE

1. A Windows machine is being used to showcase the use case where Selenium Grid, Appium node instances, and Genymotion AVDs will be hosted.
2. Genymotion Android Virtual Devices (AVDs) will be used to showcase the use case.
3. Appium app installed from binaries downloaded from [Appium Installer](https://bitbucket.org/appium/appium.app/downloads/AppiumForWindows_1_4_0_0.zip).
4. Android SDK is configured.
5. ADB device command lists all the virtual devices running on the same machine.

You should see something as shown below:

C:\>adb devices

List of devices attached

192.168.56.101:5555     device

192.168.56.102:5555     device

This is the command to show that we have 2 devices which are ready for use:

192.168.56.101:5555 is S4 Device and 192.168.56.102:5555 is S5 Device.

STEPS FOR CONFIGURATION

1. Hosting Selenium Grid

The objective of hosting Selenium Grid is to redirect the test script request to the correct device based on the desired capabilities set to decide which device to use for running the tests at run time. Use the following commands to host the Selenium Grid on port 4444.

java -jar selenium-server-standalone-###.jar -port 4444 -role hub

eg - java -jar selenium-server-standalone-2.47.1.jar -port 4444 -role hub

After executing this command the Selenium Grid is ready to listen for requests coming on port 4444 using the URL:

http://<Selenium\_Grid\_Host\_Machine\_Ip>:4444/wd/hub

1. Hosting 2 Appium Server Instances for 2 Mobile Devices

Now we have Selenium Grid hosted on an IP with a particular port where we can connect to. To configure 2 Running AVDs with Selenium Grid we need to create interfaces of the Appium Server. We have to run 2 instances of the Appium Server each associated with one of the Mobile devices and on another end connected to Selenium Grid.

SETTING UP AN APPIUM NODE INSTANCE FOR ONE DEVICE (S4) TO REGISTER WITH SELENIUM GRID

1. Creating the Appium node configuration for the S4 Device (appium\_node\_S4.json) using similar contents as shown below in the JSON file –

{  
"capabilities":  
     [  
       {  
          "deviceName": "**192.168.56.101:5555**",  
         "version":"4.4.2",  
         "maxInstances": 3,  
         "platformName":"ANDROID"  
       }  
     ],  
"configuration":  
{  
   "cleanUpCycle":2000,  
   "timeout":30000,  
   "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",  
   "url":"http://**20.20.4.71:4723**/wd/hub",  
   "host": "**20.20.4.71**",  
   "port": **4723**,  
   "maxSession": 6,  
   "register": true,  
   "registerCycle": 5000,  
   "hubPort": 4444,  
   "hubHost": "**20.20.4.71**"  
}  
}

*Note: devicename is one of the keys of the Appium node which we are configuring. Here the JSON describes that the Appium node will be associated with the S4 device and the device will be listening for commands coming on the Appium node server address (URL key in JSON) on port 4723.*

On the other side, hubPort and hubHost parameters decide which port and host machine Selenium Grid is hosted on and in turn registers the Appium node with it.

**[See Also:**[**Mobile Test Automation Using Robot Framework**](http://blog.zymr.com/mobile-test-automation-using-robot-framework)**]**

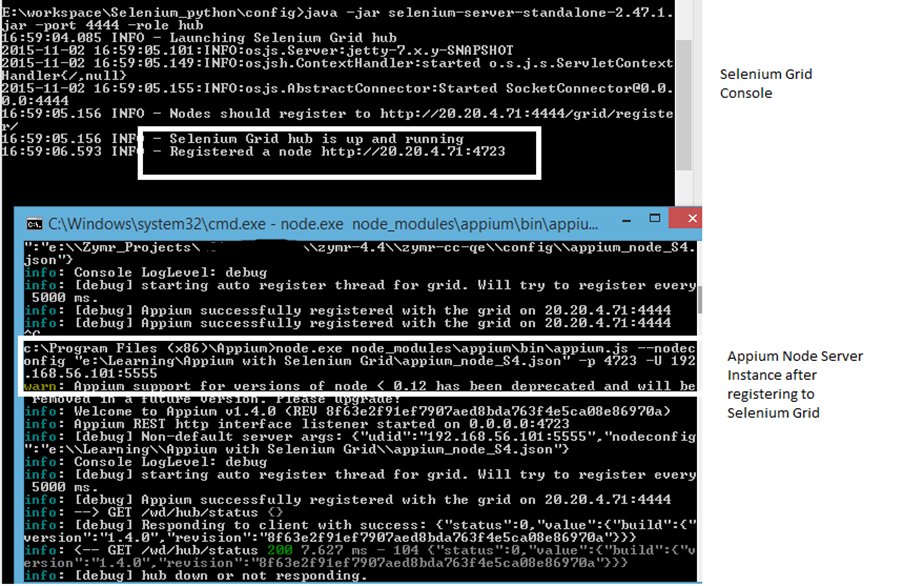
1. Open Command prompt and navigate to the location where Appium was installed. Run the following command to host the Appium node instance:

node.**exe node\_modules\appium\bin\appium.js --nodeconfig <path to appium\_node\_S4.json> -p 4723 –U 192.168.56.101:5555**

*Note: This command explains to the host Appium node server instance for devices which have UUID (192.168.56.101:5555) with the parameter –U and port 4723.*

The Selenium grid http://<Selenium\_Grid\_Host\_Machine\_Ip>:4444/wd/hub will communicate with the Appium server instance on[http://**20.20.4.71:4723**/wd/hub](http://20.20.4.71:4723/wd/hub) and in turn will reach out to the S4 device with the REST protocols.

Once this is up and running you will see the notification in the Selenium grid console as an Appium node is registered as shown below:

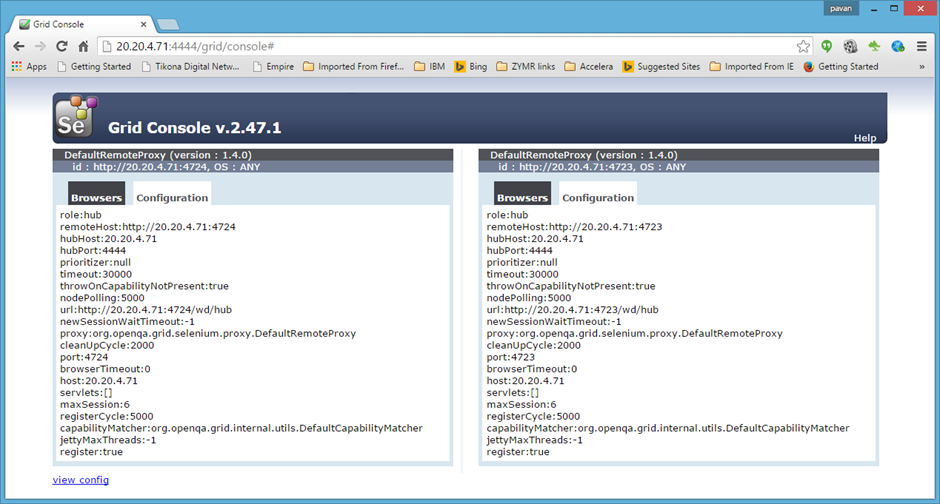


In the same way create the Appium node JSON configuration file for the S5 device as shown below and host another instance of the Appium node server -

{  
"capabilities":  
     [  
       {  
         "deviceName": "**192.168.56.102:5555**",  
         "version":"4.4.2",  
          "maxInstances": 3,  
         "platformName":"ANDROID"  
       }  
     ],  
"configuration":  
{  
   "cleanUpCycle":2000,  
   "timeout":30000,  
   "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",  
   "url":"http://**20.20.4.71:4724**/wd/hub",  
   "host": "**20.20.4.71**",  
   "port": **4724**,  
   "maxSession": 6,  
   "register": true,  
   "registerCycle": 5000,  
   "hubPort": 4444,  
   "hubHost": "**20.20.4.71**"  
}  
}

*Note: When hosting another instance of the Appium Node server for the S5 device we need to host it on a port other than the one that was used by the Appium server previously hosted for the S4 device i.e. 4723. So we choose 4724 as the port to host the Appium server for the S5 device and changed the device name to****192.168.56.102:5555****which the S5 device is running.*

Check the Selenium Grid Console to see if both the nodes are registered correctly similar to the snapshot shown below:



Now both the Appium nodes are registered successfully to the Selenium Grid and ready to work with the Test Automation script.

SETTING THE DESIRED CAPABILITY IN THE TEST AUTOMATION SCRIPT

Regardless of which programming language library we choose to write automation scripts, we need to set the desired capabilities for the Remote Webdriver. For e.g. let’s say we want to execute a script to launch a default email app on the S4 device. We will set the desired capabilities as shown below in the automation script.

desired\_caps[*'platformName'*] = *'ANDROID'*   desired\_caps[*'platformVersion'*] = *'4.4.2'*   **desired\_caps[*'deviceName'*] = *'192.168.56.101:5555'***desired\_caps[*'appPackage'*] = *'com.android.email'*     appactivity = *'.activity.Welcome'*     desired\_caps[*'appActivity'*] = appactivity  
driver = webdriver.Remote(*'http://20.20.4.71:4444/wd/hub'*,  
                                         desired\_caps)

We have set deviceName = '192.168.56.101:5555' which is the UUID of the S4 device in our scenario and our scripts are sending requests to the Selenium Grid instance hosted remotely on [***http://20.20.4.71:4444/wd/hub***](http://20.20.4.71:4444/wd/hub) which is hosted on the same machine. If Selenium Grid is hosted on any other machine then the respective URL with the correct hostname/IP address should be passed to the Remote Webdriver. The requests coming to this instance will be redirected to the respective Appium node instance whose desired capability matches with the desired capability set in the script.

Similarly, if we need to launch an email app on the S5 device only one parameter needs to be changed in the script as shown below. Requests will be redirected to the S5 device by Selenium Grid to the respective Appium node server instance of the S5 device:

desired\_caps[*'deviceName'*] = *'192.168.56.102:5555'*

**Everything you need to know about outsourcing technology development**Access a special Introduction Package with everything you want to know about outsourcing your technology development. How should you evaluate a partner? What components of your solution that are suitable to be handed off to a partner? These answers and more below.

[Introduction Package](http://cta-service-cms2.hubspot.com/ctas/v2/public/cs/c/?cta_guid=bfcaa293-8a16-47e7-a883-3fd5d0af2d3b&placement_guid=50ef5648-7149-404e-aceb-a1e3993be159&portal_id=429321&redirect_url=APefjpFGYYQiQetUvKPCHzY17Irje_L7-1UCgsJs0_SIHIMAplD6hudkd7cEDK5quCR-V2w0C3reiQ6OqOXSEJjwFT-BRDuyR-HLvtrCa7k21GESkIoETr1k5qHa7xDOaUMB7hSruj2z_9qOi2XTz8jnfsXT3mqXivy-M-0ZydccuYY9-zU1dZzEuN2hbLRCvIdkQwD_QqWvvX9Iy8eiMJ1JD4eSIRrVNLXgUSKixX0PTg95Y53-WglBcc-zO_nRCyibPzExrxR4aDx6CBuiwyd1AkYgZNNSYw&hsutk=74e8ec03359b156cc67d93693be5bb25&utm_referrer=http%3A%2F%2Fblog.zymr.com%2Fjenkins-integration-with-ios-2&canon=http%3A%2F%2Fblog.zymr.com%2Fjenkins-integration-with-ios-2&pageId=3264805476&__hstc=123617428.79ed5eeb736f16e876aa885e84d1d718.1477378979532.1477378979532.1477378979532.1&__hssc=123617428.1.1477378979533&__hsfp=940351450)

**Blackbern dude**

12/21/2015, 3:13:41 AM

Hi, thank you for your manual. But it is not working when I am trying to start command from C:Program Files (x86)Appium:   
C:Program Files (x86)Appium>node.exe node\_modulesappiuminappium.js -nodeconfig c:LIBSgrid.json -p 4723   
Console types error: appium.js: error: Unrecognized arguments: -nodeconfig c:LIBSgrid.json.   
I just copied settings above   
{   
"capabilities":   
[   
{   
"deviceName": "192.168.56.101:5555",   
"version":"4.4.2",   
"maxInstances": 3,   
"platformName":"ANDROID"   
}   
]   
"configuration":   
{   
"cleanUpCycle":2000,   
"timeout":30000,   
"proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",   
"url":"http://20.20.4.71:4723/wd/hub",   
"host": "20.20.4.71",   
"port": 4723,   
"maxSession": 6,   
"register": true,   
"registerCycle": 5000,   
"hubPort": 4444,   
"hubHost": "20.20.4.71"   
}   
}

Reply to *Blackbern dude*

**Pavan Onkar**

1/17/2016, 11:15:03 PM

Use --nodeconfig instead of -nodeconfig

Reply to *Pavan Onkar*

**Rohal Gala**

2/3/2016, 8:53:07 AM

I am raising hub using following command   
i.e java -jar selenium-server-standalone-2.47.1.jar -role hub   
what command should i add in above line to increase appium session

Reply to *Rohal Gala*

**Pavan Onkar**

2/9/2016, 12:29:42 AM

Please use maxSession=# as argument in same command , which will open that number of allowed sessions on Grid

Reply to *Pavan Onkar*

**Vadim Kovrizhkin**

2/6/2016, 3:48:46 AM

What is the command for mac?   
node.exe node\_modulesappiuminappium.js –nodeconfig -p 4723 –U 192.168.56.101:5555   
  
And   
you have mistake   
  
{   
"capabilities":   
[   
{   
"deviceName": "192.168.56.101:5555",   
"version":"4.4.2",   
"maxInstances": 3,   
"platformName":"ANDROID"   
}   
], //coma here   
...   
...   
...   
}   
}

Reply to *Vadim Kovrizhkin*

**Pavan Onkar**

2/9/2016, 12:38:58 AM

Hi Vadim,   
  
Thanks for pointing mistake and will surely fix it in article ...   
Regarding the mac thing I would get back to you soon once I try myself on the same.

Reply to *Pavan Onkar*

**Ethan Nguyen**

2/17/2016, 10:08:47 AM

I use wd js library by admc. I connect 2 nodes to the hub (selenium 2.52.0) successfully however, I got the below error:   
  
15:25:07.147 INFO - Got a request to create a new session: Capabilities [{appium-version=1.4.13, platformVersion=5.1, browserName=chrome, platformName=ANDROID, deviceName=097637cf2db8}]   
2016-02-17 15:25:07.218:WARN:osjs.ServletHandler:qtp1484594489-15: /wd/hub/session   
java.io.IOException: java.lang.IllegalStateException: Not a JSON Object: null   
  
Have you come across this error before? Any suggestion. Thanks   
  
p/s: for wdjs i set up   
driver = wd.promiseChainRemote(serverConfig)   
I run these test beofore directly with Appium.

Reply to *Ethan Nguyen*

**André Mendes**

2/18/2016, 1:06:12 PM

When i ran node.exe node\_modulesappiuminappium.js --nodeconfig -p 4723 –U 0429058934 i got: Unrecognized arguments: -u 0429058934   
  
I an using a physical device, how do i do it?

Reply to *André Mendes*

**Pavan Onkar**

3/3/2016, 11:18:27 PM

I assume even you might have tried with -udid instead of -U , if yes then check if your devices are listed in "adb devices" where your appium is supposed to be hosted.   
For eg   
If on single machine you have "adb devices" command listed as   
  
> adb devices   
0429058934 device   
  
Then it will be recognized

Reply to *Pavan Onkar*

**Pavan Onkar**

3/3/2016, 11:29:24 PM

Hi,   
  
Just I recollected from Appium's official site where in latest version that have depricated the parameter -U and have kept only --udid as parameter

Reply to *Pavan Onkar*

**Petr Beneš**

2/25/2016, 1:37:40 AM

Great post, thx. Very helpfull. I just had to omit this part "–U 192.168.56.101:5555" in the command otherwise I got error "appium.js: error: Unrecognized arguments: -U 192.168.56.101:5555"

Reply to *Petr Beneš*

**Petr Beneš**

2/25/2016, 1:45:44 AM

Nevermind, I just found the error. It doesnt work if you ctrl+c ctrl+v command from the page, because there is different symbol in front of U: "node.exe node\_modulesappiuminappium.js --nodeconfig -p 4723 –U 192.168.56.101:5555" You see?

Reply to *Petr Beneš*

**Radek Bokuvka**

4/19/2016, 3:26:25 AM

Thanks much Peter for pointing out about the copy-paste issue.

Reply to *Radek Bokuvka*

**Pavan Onkar**

3/4/2016, 12:29:54 AM

Thanks Peter, for going through the post.   
Sorry for late reply.. Will keep headsup on the queries coming furtther

Reply to *Pavan Onkar*

**FX Gentilhomme**

9/29/2016, 9:38:58 AM

Hi!   
Thank you for this article, it's very usefull.   
I'm trying to do quite the same configuration, but I'm facing issues configuring the appium with real iOS devices. I'm not sure of which deviceName is should use, and if the udid should replaced it, but I have an invalid param when appium try to register on the selenium hub.   
If you can give me a hand on this, I will be really thankfull. I've created a stackoverflow issue for that (http://stackoverflow.com/questions/39727539/how-to-configure-nightwatch-js-selenium-hub-appium-real-mobile-devices)