

QA Automation using Selenium Server

We should be using selenium (hub + nodes) instead of our current approach of running nodejs stack on windows to handle automation. This document is good source of knowledge on how to set up selenium automation.

Downloading the server

The quickest way to download the selenium server is through SeleniumHQ - download the stand alone server:

<http://www.seleniumhq.org/download/>

Downloading Drivers and Registering them to server

Now you need to download various drivers and register them to selenium server:

For IE, you can download from:

- <http://www.seleniumhq.org/download/>
- <https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver>

For Chrome:

- <https://sites.google.com/a/chromium.org/chromedriver/downloads>

Setting up VMs to run selenium server

You need to first run selenium server with chrome and ie driver path to support running it against chrome/ie. You can use this windows bat as a starting point:

```
java -Dwebdriver.ie.driver=C:\automation\IEDriverServer.exe  
-Dwebdriver.chrome.driver=C:\automation\chromedriver_win32\chromedriver.exe -jar  
.\selenium-server-standalone-2.53.1.jar
```

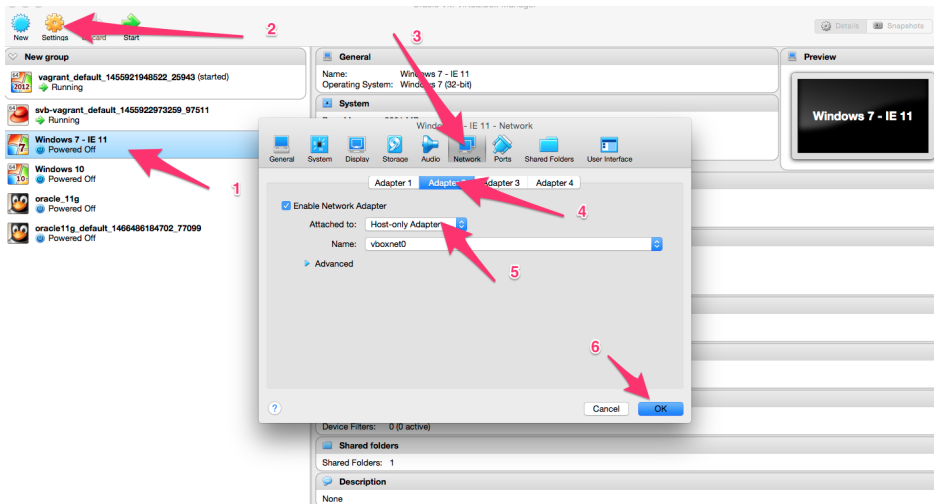
Setting up VM to talk to host machine w/ virtualbox

You are running windows guest and running the server from osx (host), you need to find out the ip address that windows can use to talk to the host machine. You can figure this out using ifconfig:

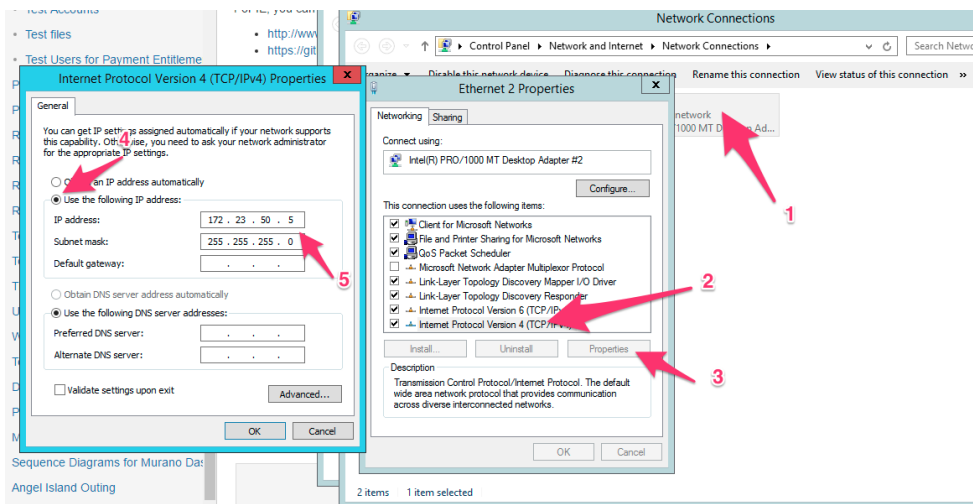
```
$ ifconfig | grep inet  
inet6 ::1 prefixlen 128  
inet 127.0.0.1 netmask 0xff000000  
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1  
inet6 fe80::f65c:89ff:fe8f:2ec1%en0 prefixlen 64 scopeid 0x4  
inet 10.11.85.74 netmask 0xfffffc00 broadcast 10.11.87.255  
inet6 fe80::d02d:79ff:fe8f:b45b%awdl0 prefixlen 64 scopeid 0x8  
inet 172.23.50.1 netmask 0xffffffff broadcast 172.23.50.255  
inet 10.136.2.78 --> 10.136.2.78 netmask 0xfffffc00
```

In this example, the host box could be accessed using any of the listed ip, and for muno services `172.23.50.1` is the shared host-only network between osx host and windows guest machine.

You may also need to configure windows guest machine by adding additional 'host-only' network card:



Inside windows guest machine - you may have to configure the network card to use static ip address with the correct submask:



Configuring protractor to run against selenium server

You can modify `protractor.conf` and set seleniumAddress field to the server that's running the selenium server.

```
exports.config = {
  allScriptsTimeout: 11000,
  rootElement: 'html',
  //seleniumServerJar:
  './node_modules/protractor/selenium/selenium-server-standalone-2.48.2.jar',
  seleniumAddress: 'http://172.23.50.5:4444/wd/hub',
}
```

Running protractor against windows ie, chrome, firefox, etc

Right now payment protractor scripts is set up to take in the address where the payment web server is running. You can pass in selenium host address to specify which selenium server to run against, as well and browsers.

Here's some examples on command line arguments:

```
## Running against mock server running on host against windows on 172.23.50.5, running ie
npm run mock-protractor -- \
  --seleniumAddress='http://172.23.50.5:4444/wd/hub' \
  --params.browsers='ie' \
  --params.baseUrl='http://172.23.50.1:9002/'

## Running against mock server running on host against windows on 172.23.50.5, running chrome
npm run mock-protractor -- \
  --seleniumAddress='http://172.23.50.5:4444/wd/hub' \
  --params.browsers='chrome' \
  --params.baseUrl='http://172.23.50.1:9002/'

## Running against mock server running in a vagrant box
npm run mock-protractor -- \
  --seleniumAddress='http://172.23.50.5:4444/wd/hub' \
  --params.browsers='chrome' \
  --params.loginUrl='http://172.23.50.4/auth/' \
  --params.baseUrl='http://172.23.50.4/payment/'

## Running against real server running in vagrant box - against windows ie
npm run protractor -- \
  --seleniumAddress='http://172.23.50.5:4444/wd/hub' \
  --params.browsers='ie' \
  --params.loginUrl='http://172.23.50.4/auth/' \
  --params.baseUrl='http://172.23.50.4/payment/'

## Running against qa server - against windows ie
npm run protractor -- \
  --seleniumAddress='http://172.23.50.5:4444/wd/hub' \
  --params.browsers='ie' \
  --params.loginUrl='https://qa4.svbconnect.com/auth/' \
  --params.baseUrl='https://murano.qa4.svbconnect.com/payment/'
```

Quick Start

You can get Started by using this zip file and extracting it to `C:\` drive.



automation.zip