QA BrowserStack Evaluation: PoC Plan

Wednesday, January 24, 2018 9:49 AM

1. Introduction

The company has been evaluating a replacement for selenium grid and have picked BrowserStack to evaluate. Mobile needs to be part of the evaluation since we have automation test running on the grid.

This is a plan for the Proof of Concept for the QA Evaluation of BrowserStack. This PoC is the technical verification part of QA's evaluation of BrowserStack. (Other parts of the evaluation might include business perspectives, for example.)

The goal of the PoC is to verify BrowserStack against our requirements. Main areas of interest includes possible solutions to some existing shortfalls or pain points with our current system, such as image diff failures due to subtle pixel differences in our grid environment, and mobile (Appium) test startup getting contention sometimes. It will also verify that BrowserStack works with specific pieces of FactSet technology and applications, such as FDSAv2 overrides. We will also evaluate its performance and robustness. The purchasing decisions might take the results of this PoC into consideration.

BrowserStack website: https://www.browserstack.com/

RPDs:
RPD:36870484 Preparation for BrowserStack POC - Action items for Tellus Framework and Scripting team
RPD:37432435 Evaluating browserstack mobile device testing
RPD:34302004 Browserstack Evaluation Firewall Needs
RPD:36649993 Access to bstackdeva01.pc.factset.com for BrowserStack connection and testing

2. Preparation Phase

There will be a preparation phase to get the equipment ready for testing, and to verify that testing with BrowserStack can access on internal (e.g. staging) URLs. As BrowserStack accesses internal apps through BrowserStackLocal.exe ("local testing" in BrowserStack terminology), we need a server with access limited to specific whitelisted URLs; it will run BrowserStackLocal.exe. The equipment will be listed in the Equipment section below.

Tasks:

Item	Assigned To	Status	Comments
Setup BrowserStackLocal.exe server	Windows Team	**Completed**	bstackdeva01 server
Whitelist internal URLs on BrowserStackLocal.exe server	Security Team, Bhupender, Lilun	**In Progress**	Need to collect all internal URLs from scripters for each application. We noticed that partial pages are loaded for some urls on bstackdeva01, external links may need to be added to the whitelist if any. More testing is needed. **=> TODO: use fiddler/wireshark to get list of URLs**
iOS test script	Swammy	**Completed**	
Android test script	Swammy	**Completed**	Swammy provides simple Android QA test scripts. Note: There is currently no full QA test script available.
Query our current concurrency stats: the max number of concurrent Selenium Grid tests (for load test and purchasing decision)	Joseph	**Completed**	
Create load test script	Lilun		

Selenium Grid Concurrency Stats

Tellus current Selenium Grid's peak concurrency is shown in the table below. Non-iOS browsers are Windows platform.

Selenium Grid Concurrency Stats Since 1/1/2017:

Browser	Peak
chrome	49
firefox	11
ie	3
iOS	14
Overall	49

It shows that iOS had a peak of 14, but we expected only up to 4 (the max number of devices). It is probably related to unexpected long job durations of more than one hour around 1/18/2018, as shown by data in the database, while typical job durations where 1 to 15 minutes. The cause of that "aberration" was probably due to jobs getting "hung" (which could be scripting or system issue).

If you screen out the aberrations, then the query below shows that iOS has peak concurrency of 3 from 1/1/2018 to 1/17/2018 00:00:00. These would be more "normal" statistics.
Concurrency Stats from 1/1/2018 to 1/17/2018 00:00:00:

Browser	Peak
chrome	22
firefox	0
ie	1
iOS	**3**
Overall	22

Preparatory Tests:

Test Item	Status	Assigned To	Comments
Web - Verify that BrowserStack test can access internal URLs	**Passed**	Lilun	Tested with generic Ruby Selenium test against staging URL, through BrowserStackLocal.exe on tellus server with slight securi ty requirement. More internal URLs need to be added to the whitelist on bstackdeva01.
iOS - Verify that iOS app can run in BrowserStack , including file transfer of app package	**Testing**	Lilun	Tested with FactSet app (App Store build).
Android - Verify that Android app can run in BrowserStack, including file transfer of app package	**Passed**	Lilun	Tested a sample apk provided by BrowserStack with public URLs. It worked fine.
Verify that BroserStackLocal.exe can use proxy	**Passed**	Lilun	These BroserStackLocal.exe options worked with BrowserMob Proxy server: `--local-proxy-host 127.0.0.1 --local-proxy-port 8888 --force-proxy`

3. Evaluation Criteria

The following needs to be evaluated in the QA PoC phase:

(1) Availability
Evaluation: BrowserStack should be available at any time we do testing.
Notes:
i. BrowserStack Web site claims 99.99% uptime.
ii. According to BrowserStack representative, BrowserStack has infrequent maintenance downtime, about twice a year, usually early Sunday mornings. It has fallback servers. Please see meeting notes (OneNote on H: drive): Meeting with Browser Stack Team

(2) Capability
BrowserStack should be able to run our test scripts.
Evaluation: Test three major FactSet apps to verify this: IOS app, FDSAv2 Overrides apps (Universal Screening), Android app (one sample script and a sample FactSet script).

(3) Accessibility
BrowserStack should be able to access our internal apps to do local testing.
Evaluation: Run major FactSet apps to verify this.

(4) Stability
BrowserStack should run the scripts in a continuous way.
Evaluation: Run major FactSet App scripts to verify this.

(5) Consistency
Test results should be consistent.
Evaluation: Run major FactSet app scripts multiple times to verify this.

(6) Performance
BrowserStack should provide consistent performance.
Evaluation: Run major FactSet app scripts multiple times to see whether the performance is similar.
Section 3.2 below.
- Max number of concurrent tests
- How does BrowserStack handle number of tests exceeding max purchased "parallel tests"

The items required for this evaluation to pass is indicated by "Required" below. Non-required items are "nice to have", but optional with respect to inclusion in this PoC. The criteria include specific pieces of technology that need to be proven to work.

3.1 Test Items:

Test/Purpose	Required?	Application	Test Script	Assigned To	Status	Comments
Web App - FDSAv2 overrides **Purpose**: BrowserStack works with FDSAv2 overrides, internal URL	Yes	Universal Screening	qa-test-us (version 223.0.0)	Lilun	**Passed**	Pramod confirmed that FDSAv2 overrides were working fine with BrowserStack.
iOS native app (app store build) running against production url[2] **Purpose**: iOS app works with production URL	Yes	FactSet iOS app	Lilun's custom Script (based on Swamy's)[1]	Lilun	**Passed**	There were some errors with driver initialization. Copied the driver initialization part from the working example of BrowserStack, and it started working.
~~iOS native app (snapshot build)~~ ~~**Purpose**: iOS running with lima staging and LIMA override (wireless_devel)~~[2]	~~No~~	~~FactSet iOS app~~	~~Script from Swamy~~[1]	~~Lilun~~	~~Testing~~	Dropped, see RPD Comment 37835714
iOS - robustness **Purpose**: Consistently, successfully start test every time (repeated consecutive tests)	Yes	FactSet iOS app	Lilun's custom Script (based on Swamy's)[1]	Lilun	Passed	Tested each of two scripts more than 10 times.
Android mobile 3.0 hybrid web app **Purpose**: Android running with staging/internal URL[2]	Yes	FactSet Android app	Test script in OneNote on H: drive: Android SDK and Appium on new Tellus servers for Selenium grid	Lilun	Testing	The fds login page blocked issue has been fixed. Got a different error. Will work with Swamy to fix it.
Web - Load test **Purpose**: a) Should get at least comparable performance as Tellus Selenium Grid b) CPU% should stay moderate on BrowserStackLocal.exe host c) BrowserStack should gracefully handle case when #requests exceed max purchased number of parallel tests ("hold queue")	Yes	Universal Screening	qa-test-us (version 223.0.0)	Lilun		Test potential BrowserStackLocal.exe bottleneck.
Image diff testing	Yes					We will leave this test to Josh Fee --- as this is a major concern of Engineering.

[1]Scripts are from zip file, which we will archive in known location shortly
[2]According to Comment (Kun Lu; 1/25/2018 02:51 PM (GMT))

LIMA overrides of Web applications will *not* be included in this PoC, as the number of test cases that uses LIMA is relatively small, and they are migrating off LIMA by the end of FY18, at the latest:

Project	#Test Cases	Timeline to migrate off LIMA
AT4	299	End of Jan
QS	70	End of Jan
PS3	23	End of Jan
BPM2	97	End of this FY18
AXP2	48	End of this FY18

Please see: Application List with Lima Overrides Requirement

Note: Testing with LIMA overrides may require a proxy server, which is substantial administrative and maintenance effort.

For Android app, we will test with simple test script (with a few elements), as there is currently no existing full QA test script.

3.2 Performance Metrics

Test Product	Test Case	Tellus Avg	Tellus Max	BrowserStack	% Improvement

		Duration (Minutes)	Duration (Minutes)	Duration (Minutes)	(compared to Tellus average)
qa-test-us	Demo_Workflow	8	59	10	-25%

4. Equipment and Resources

a. Equipment

i. bstackdeva01: BrowserStackLocal.exe host (BrowserStack traffic accesses internal URLs through it)
ii. Whitelist URLs on bstackdeva01: List of app urls

a. Software Tools

i. BrowserStackLocal.exe binary is delivered by BrowserStack
ii. Load test tool will be Script to launch multiple Selenium tests at the same time

5. Applications under Test and Test Script

Application under Test	App Location	Test Script Location
Universal Screening	https://universalscreening.apps.factset.com/#/?QA_MODE=1	qa-test-us (version 223.0.0)
FactSet iOS App	• App Store FactSet.ip • Release FactSet.ipa	Test script is from Swammy
Android App	App package is attached to OneNote on H: drive: Android SDK and Appium on new Tellus servers for Selenium grid	Test script in OneNote on H: drive: Android SDK and Appium on new Tellus servers for Selenium grid

6. Personnel

QA Personnel

Personnel	Role
Lilun Cao	Owner of project, testing with BrowserStack
Bhupender Agarwal	Help with planning, test scripts, and other questions
Krishnateja Surapaneni	Help with test scripts and other questions
Joseph Hwang	Help with planning, troubleshooting
Mahendra Tribhuvan Sura	Help with providing test scripts
Swammy Kommu	Provide iOS and Android test scripts
Pramod Devi	Help with qa-test-us test script

Engineering Contacts

Personnel	Role
Josh Fee	Engineering BrowserStack PoC
Kun Lu	iOS/Android Engineering

7. Schedule

Tentatively, the PoC testing will last 2 weeks.

Event	Date	Comments
Start of PoC	2/5/2018	
End of PoC	2/16/2018 [Tentative]	

8. Exit Criteria

When all required evaluation criteria have completed, this PoC may conclude. The required Test Items under Evaluation Criteria must all pass for the PoC to pass.

9. Risk and Contingency

We have verified that BrowserStack "local testing" (BrowserStackLocal.exe) can access internal URLs, and generic Android with public URLs run in BrowserStack. We are still testing iOS apps with BrowserStack. However, application execution might reveal additional URLs being blocked on bstackdeva01, as it progresses farther, so prompt response to additional whitelist will be essential.

Since we only tested some tests for some applications, there may be some potential issues with some applications when running with BrowserStack.

Note: There are known issues with Appium, such as the inability to test drag and drop (but some workaround might exist). Those types of issues are not limited to BrowserStack.

In the preparatory testing, we have verified that BrowserStackLocal.exe works with local proxy (BrowserMob Proxy), but we have not attempted to inject headers, as we currently do not require LIMA overrides. However, if we need LIMA in the future, a proxy server would be needed for header injection. It would require setup and administration effort.

If we encounter issues during the test, it might take additional time to troubleshoot, which might delay the completion of the testing, unless we can flexibly increase resources for that.

Reference

- Consolidated Evaluation Criteria: https://gist.github.factset.com/ifee/62a63a0395e7b66782842121e47c288f
- BrowserStackLocal.exe: https://www.browserstack.com/local-testing

Application List with Lima Overrides Requirement

Friday, January 12, 2018 8:25 PM

Platform

S.No.	Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
1	FDSChartJS	NA			-
2	Formatter	NA			-
3	Styleset	NA			-
4	DataCheck	No	No	Yes	-
5	FactSearch	No	No	Yes	-
6	Notify	No	No	No	-
7	File Manager	No			-
8	Market Intelligence	No			-
9	Online Assistant (OA3)	No	Yes	Yes	-
10	ID-widget Commingled	No	No		-
11	Identifier Lookup(IL4)	No	No		-
12	ID Widget Angular	No	No		-
13	Mobile	NA			-
14	HTML Data Downloading	NA			-
15	JS Active Graph	NA			-
16	File Dialogs	No	No	No	-
17	Web Formula Lookup	No	No	No	-
18	LoginLogoutFactSet	No	No		-

Research

S.No	Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
1	Interactive Charthing	No	Yes	Yes	
2	Estimates	No	Yes	Yes	
3	Financials	No	Yes	Yes	
4	Overviews	No	Yes	Yes	
5	Price reports	Yes	Yes	Yes	This is SoapUI Test Case for which we need both Lima and FDSAv2
6	Economics	No	Yes	Yes	
7	Filings	No	Yes	Yes	
8	Industry	No	Yes	Yes	
9	Full Quote	No	Yes	Yes	
10	Real Time Markets	No	Yes	Yes	
11	News	No	Yes	Yes	
12	Markets	No	Yes	Yes	
13	Ownership	No	Yes	Yes	
14	Investment Research 2.0	No	Yes	Yes	
15	Idea Screening	No	Yes	Yes	
16	NextGen Screening	No	Yes	Yes	
17	Callstreet	No	Yes	Yes	
18	CodeRed	NA	NA	NA	
19	IRN	No	Yes	Yes	
20	Partners	No	Yes	Yes	

PMT Applications:

S.No	Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
1	Portfolio View	NA	NA	NA	

NOTE: This app are workstation dependent. Hence, this apps are not testable in selenium-grid environment.

Wealth Applications:

S.No	Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
1	Security Analysis	No	NA	NA	
2	Security Overviews	No	NA	NA	
3	Options & Derivatives	NA	NA	NA	

NOTE: All this apps are workstation dependent. Hence, this apps are not testable in selenium-grid environment.

CTS:

S.No	Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
1	Exchange Datafeed	No			
2	OnDemand Service	No			
3	Cornerstone	No			

Analytics:

Application Name	Lima Overrides Required (Yes/No/NA)	FDSAv2 Overrides Required (Yes/No/NA)	Does app work on public URL like <app_name>.apps.factset.com? (Yes/No/NA)	If Yes, any plans to move to get rid of this dependency? If Yes, what's the timeline?
AT4, QS, PS3	Yes	Yes	No	January end
BPM2, AXP2	Yes	Yes	No	End of this fiscal year
PA3	No	Yes	Yes	
PLM	No	Yes	Yes	
PRU	No	Yes	Yes	
AA	No	Yes		
SPAR3	No	Yes	Yes	
PDB2	No	Yes		
VRS	No	No	NA	

Project	#Test Cases	Timeline to migrate off LIMA
at4	299	End of Jan
qs	70	End of Jan
ps3	23	End of Jan
bpm2	97	End of this FY18
axp2	48	End of this FY18

List of app urls

Tuesday, December 19, 2017 7:35 PM

@JoshMoorhouse

Below are the end points that we use today for QA testing for web apps in staging and production and would need to be whitelisted for BS

Also end point from Platform engineerings are listed as well

Platform applications:

- launch.factset.com
- login-staging.factset.com
- filedialog.staging-cauth.factset.com
- formula-lookup-demo.staging-cauth.factset.com
- fdschartjs.factset.com
- oa.apps.factset.com
- identifier-lookup-demo.staging-cauth.factset.com
- notifyqa.staging-cauth.factset.com
- outlook.pc.factset.com/owa/fds_auds_testers@factset.com
- idwidgetangulardemo.staging-cauth.factset.com
- newsapps-beta.apps.factset.com
- today.apps.factset.com
- company-security-beta.apps.factset.com

Analytics:

- alphatesting.staging-cauth.factset.com
- bpm.staging-cauth.factset.com
- quantsuite.staging-cauth.factset.com
- axp.staging-cauth.factset.com
- portfolio-dashboard.staging-cauth.factset.com
- bcm.staging-cauth.factset.com
- portfoliosimulation.staging-cauth.factset.com
- vrsqa01.open.factset.com
- quant-model-manager.staging-cauth.factset.com
- pint.staging-cauth.factset.com
- audit-inspector.staging-cauth.factset.com
- audit-inspector.factset.com
- pru.apps.factset.com
- portfoliolistmanager.factset.com
- peru.apps.factset.com
- spar.staging-cauth.factset.com
- portfoliolistmanager.staging-cauth.factset.com
- peru.staging-cauth.factset.com
- pru.staging-cauth.factset.com
- pint.inhouse-cauth.factset.com
- pa.apps.factset.com
- pa.staging-cauth.factset.com

- securitytradingutility.staging-cauth.factset.com
- securitytradingutility.apps.factset.com
- alphatesting.factset.com
- bpm.apps.factset.com
- quantsuite.factset.com
- axp.apps.factset.com
- portfolio-dashboard.apps.factset.com
- acm.staging-cauth.factset.com
- portfoliosimulation.factset.com
- quant-model-manager.factset.com

Research:

- filings.apps.factset.com
- dashboard.apps.factset.com
- universalscreening.apps.factset.com
- markets-beta.apps.factset.com
- company-security.apps.factset.com
- callstreet.staging-cauth.factset.com
- markets.apps.factset.com
- economics.apps.factset.com
- ideascreening.staging-cauth.factset.com
- industry.apps.factset.com
- irn.staging-cauth.factset.com
- irn.inhouse-cauth.factset.com
- company-security-beta.apps.factset.com
- ownership.apps.factset.com
- contact-screening.apps.factset.com
- views.apps.factset.com
- newsapps-beta.apps.factset.com
- cmreports.staging-cauth.factset.com

Platform Engineering

ewolbachpc.pc.factset.com
jfeepc.pc.factset.com
ddivechapc.pc.factset.com
ejohnson02pc.pc.factset.com
arichardspc.pc.factset.com
ahinespc.pc.factset.com
lcerritopc.pc.factset.com
rgannampc.pc.factset.com
khwangpc.pc.factset.com
tcrescenzipc.pc.factset.com
rkirschnerpc.pc.factset.com
htroegerpc.pc.factset.com
mdavispc.pc.factset.com
auppalpc.pc.factset.com
unixdeva19
unixdeva22
unixdeva16
jenkinsbuilddeva03
All domains ports 9876-9886
fdschartjs.factset.com:80

Additional URLs:
cdn.factset.com
login.factset.com
login-inhouse.factset.com
fds-mobile-login.factset.io

1/26/2018:
rtgateway-ih-vip.prod.factset.com
financials.apps.factset.com
xd-connect.apps.factset.com
public-snapshot.apps.factset.com
factsearch.factset.com

From <http://is.factset.com/rpd/Summary.aspx?messageId=34302004>

Meeting Notes

Thursday, January 25, 2018 2:31 PM

See sub-pages.

Meeting with Browser Stack Team

Friday, December 01, 2017 11:20 PM

Meeting Subject: BrowserStack meeting - Tentative and block time
Meeting Date: 12/1/2017 10:00 PM
Location: Go to Meeting updated in email
Link to Outlook Item: *click here*
Invitation Message
Participants

- Rajul Shrivastava (Meeting Organizer)
- Avanthi Rentala
- Joseph Hwang
- Lilun Cao
- Bhupender Agarwal
- Ramya Parthasarathy
- Surendra Babu Parchuru
- Vasu Nakka

Notes

- Each single driver session at BS side will generate logs which can be viewed on BS Dashboard.
- Text Logs, Console Logs and Network Logs (which are basically in the form of text) are retained for 90 days on BS servers.
- Screenshots and Videos will be stored for 30 days but you can get it later via API call and store them locally.
- Video limit
  - Can store long videos too but would be nice to have shorter videos.
- Will each test case have separate longs when running on one driver instance?
  - No, logs will be based on each driver session. Recommendation is to run one test case per driver instance
- What is the criteria to divide the no. of test cases to different browser instance?
  - BS does not control this, we can control this via script and specify how many scripts we want to divide per driver instance?
- **If we go by each driver instance for one test case how we'll be able to cope up with the limitation on concurrent BS instances which are 30 as of now??**
- **Each test case per instance would need many serial numbers which can exhaust our SN pool. What is the plan to handle this? Even if we are ready to allocate SN, is our infrastructure ready to serve the request that sooner?**
- **How we'll consolidate a number of result logs if we go by each test case per instance??**
- BS will hard stop job after 90 mins. And timeout after 90 secs if no command is received within that window.
- It is possible to get BS logging Dashboard URL which can be used to launch the dashboard directly.
- Session IDs can also be relied on to launch BS logging dashboard.
- Upload and Download are not possible on BS servers still.
  - BS launch a VM for each driver instance. They have 1000s of VMs running.

Mobile Testing

- Using API we can upload .apk files to BS server which will return you URL with hash key pointing to the uploaded file.
- Files uploaded will be stored forever and you can use it for multiple test instances.
  You can also store multiple versions of .apk files.

Joseph's Note:

Web app testing

- Can use FactSet corporate SSL root certificate
- **BS will look into Fiddler, header injection**
- Can set cookies
- Records video of session -- can handle long videos, but easier to find things in shorter ones
- BS console text log is element level (but Tellus log is test case or assertion level)
  - Can also access BS's log by REST URL (but URL has long hash key)
- **VM hard stop at 90 minutes**
  - => Recommend parallel testing
- Console log, network logs retains 90 days. Text logs 30 days.
- Only one browser runs in a VM. It is VM, not Docker container.
- Need license key per manual UI access session (up to team license limit)
- Infrequent maintenance downtime (twice a year, early Sunday morning), has fallback servers
- Use localtesting.exe for local testing
- **Cannot export to Excel** (download) on BrowserStack machine -- no access to its filesystem
  - **=> Need BrowserStack test environment to isolate this type of tests???**

Mobile app

- we upload app binaries to BS, and get bs:<hashkey> URL, put that in caps['app'] = "bs:xxxxx"
  - API to list, update, delete apps
  - **Retention, size limit???**

Lilun's Questions:

- Can we configure 90 seconds' timeout? IE is working slow. Sometimes it may need more time to respond.
- Can multiple people access the dashboard using the same credential at the same time to check different jobs/logs?

Notes:

- Start the local server "**BrowserStackLocal.exe**" to do local testing. I tested this before.
- Play video for test cases' run. Can see what happens at each step.
- Capabilities are set in the script and have more browserstack-specific attributes.
  - Example for **local testing**:
    caps = Selenium::WebDriver::Remote::Capabilities.chrome
    caps['browser'] = 'Chrome'
    caps['browser_version'] = '59.0'
    caps['os'] = 'Windows'
    caps['os_version'] = '7'
    caps['resolution'] = '1920x1080'
    caps['acceptSslCerts'] = 'true'
    caps["browserstack.debug"] = "true"
    caps['browserstack.local'] = 'true'
- Hub url looks like <http://username:accesskey@hub-cloud.browserstack.com/wd/hub>
  - wd = Watir::Browser.new(:remote, url: "http://username:accesskey@hub-cloud.browserstack.com/wd/hub", :desired_capabilities => caps)
- Set cookies or Fiddler Header injection need to be looked into for resolution. BS has no experiences with that.
- Not sure whether BS can set trusted sites in IE.

BrowserStack Go/No Go Meeting Notes 1/25/2018

Thursday, January 25, 2018 1:38 PM

- => use fiddler/wireshark to get all urls to unblock

- Availability:
  - => Need to quantify criteria
  - Add expected downtime:
  - Meeting notes: Infrequent maintenance downtime (twice a year, early Sunday morning), has fallback servers
  - Ref: Meeting with Browser Stack Team
  - Compare data to historical downtime
  - Stability metrics
  - Universal Screening test run time about same on BrowserStack as our grid
  - Run test in parallel

Draft: Evaluation and Success Criteria

Wednesday, December 27, 2017 8:24 PM

See Updates in Consolidated Evaluation Criteria:
https://gist.github.factset.com/jfee/62a63a0395e7b66782842121e47c288f

Why external cross browser grid is necessary?
A.: As more and more FactSet applications are re-written or developed to run on browser (desktop/mobile) the requirement to do web/mobile automation has increased. Thus, we have seen 4X increase in number of automated jobs executed on our home grown Selenium Grid Environment. We are expecting 2X increase in number of jobs with base as current statistics which stands at 23340 jobs per year. Also, this is just a number from testing against Chrome browser.

In order to meet the growing demand, it is very difficult to maintain the home grown selenium grid and make it highly available. Hence, in order to support the future scaling we need to evaluate external vendor, in this case Browser Stack, which specializes in providing such services.

Areas we want to evaluate:

1. Features such as live monitoring (multiple users), logs availability, video logs, availability of browser version etc.,
   We have been having a challenge to monitor what's going on when job is scheduled to run on Selenium-Grid environment. Thus, having live monitoring feature is very vital for us which will help us identify where the failures are arising from. Also, availability of different type of logs such as Video, console and BS custom logs will help us to debug the issues faster.
   Adding new browser versions to our grid system is a costly deal as one person from framework team has to ensure the compatibility with drivers, make the hub available for testing before productionizing etc. Thus, availability of latest browser versions would reduce the time teams invest to confirm if newly added browser versions are set correctly.

2. Number of Jobs we can schedule in single instance.
   Currently, we have a multi-plan feature developed which allow us to schedule whole suite of automated test cases and queue the jobs if resources are not available. Though, queuing mechanism is not from BS we want to verify if our current system can propagate jobs on BS server smoothly without any job losses.

3. Time taken to execute each job.
   Goal here is to identify how long the job takes to complete execution when it is compared with execution time on our system. If jobs complete faster on BS environment we can plan for shedding off mode for BS environment.

4. Availability of system.
   Many times due to high volume of jobs system resources are utilized completely which will make underlying system to non-responding. Thus, we want to ensure that BS environment is available all the time for utilization.

5. Stability of system.
   There are many cases we see with our system that results are not stable. If some jobs fails during first schedule it is not guaranteed to fail again at same point when scheduled again. Thus, lot of intermittent failures are seen due to unstable environment. We want to check the fault tolerance of BS environment

Success criteria for Evaluation:

1. Timeliness of result irrespective of load.
   If we are able to get results with consistent output and without delay then it'll help us reduce the noise generated due to delays in receiving results.

2. No downtime, and no environment/infrastructural issues.
   High system availability and no environment issues will help team to focus on identifying actual application issues rather than investigating script flakiness.

Additional support required:

1. Injecting headers <not_urgent>

BrowserStack Set up for Tellus

Monday, December 04, 2017 3:34 PM

BrowserStack documentation

Here is the documentation for the topics we discussed on the call:

- https://docs.google.com/a/browserstack.com/forms/d/11lIJHrJ6h7dz4Ku0_g1cdJK4D7vF39ykYXBp-WwSm7w/viewform?edit_requested=true
- Downloading files during Automate
- Local Testing Modifiers
- Selenium Capabilities
- Browsers and Devices for Automate

Infrastructure Set up and JEP set up (not PoC)

BrowserStack Set up for Tellus should work like Selenium Grid today.

For BrowserStack set up we will have to download the **BrowserStackLocal.exe on Tellus App server which act as client for SGE jobs today.** < new requirement>

Ask from Josh Moorehouse if this okay to install of Tellus server or should use a VM ??

For Test Execution Tellus would call Browserstack hub url with Service account created for Tellus execution for Browserstack

JEP will need to be modified to add BrowserStack as environment**.** < new requirement>

We should be able to same Dispatcher and parser as SGE today

BrowserStack on its side do not provide any queue or scheduling abilities.

Job Queue and scheduling will have to be managed by Tellus like it does today for SGE jobs

BrowserStack can allow for Test Plan execution with test cases using same browser instance.

1 Browser instance can live up to 90 mins to 2 hours.

Hence test cases need to be set such that they finish within 2 hours else the job will terminate

Logging and result processing

Current logging and parsing process to write back on SWP need not change for BrowserStack. It will remain same.

BrowserStack provides rich logging for 1 Browser session (1 test case or 1 Job for timing < 2 hours)

Dashboard with Text Logs for each testcase/job/Browser session

Network Logs

Console Logs

Video Logs

Screen shots

These assets like Screen shots or Videos can be downloaded via an REST API which Browserstack provides

As part of result processing we could add the link to the Browserstack Video on Email or SWP response for people to view it

For test running on BS we could add a link to the video for people to see it when needed. < new requirement>

Keep in mind only people with access to BrowserStack will have access to this video

This will avoid the need to rerun test cases in case of failures

Text Logs like Network, Console etc can live up to **90 days**

Video and Screen shot live upto **30 days**

The logs for a Job will include everything on a job as part of execution (all test cases)

A video for job will be long based on number of test cases in the job hence it not be very efficient in term of troubleshooting

Tellus Job Definition

Jobs definition for BrowserStack will not change and will remain as today

QA team need to make sure just enough test cases are running as job so that they do not run over 2 hours as it will lead to Job terminate.

May be Tellus would have to set up limitation for how many test cases can be selected in a job< New Requirement>

Manual Test and Debugging

Manual Testing when needed in order to reproduce the issue.

This will require each authorized tester (Manual and Automated) a key which they would use to log in to BS site and find the job and play video or create similar VM and run the test again

Mobile Testing

- Using API we can upload .apk files to BS server which will return you URL with hash key pointing to the uploaded file.
- Files uploaded will be stored forever and you can use it for multiple test instances.

You can also store multiple versions of .apk files.

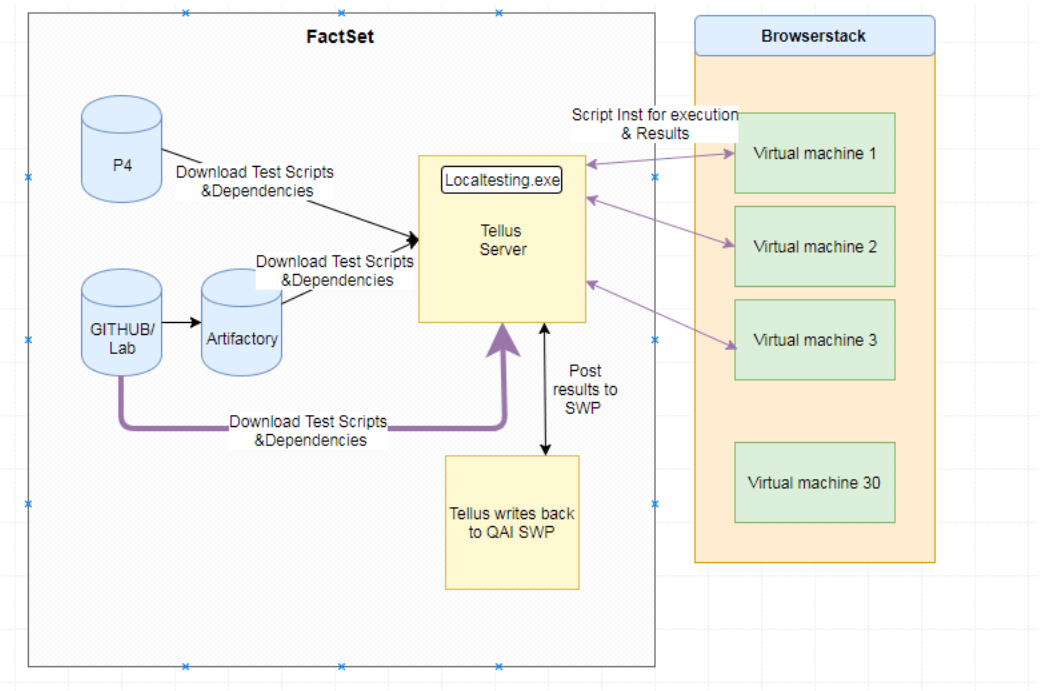
Browserstack has an API which can be hooked up to tellus to show which version of .APK file we want to use to test while submitting a job

Open Questions:

Do we need to host Local Testing.exe on Tellus?

How we will do Fiddler injection BrowserMob vs some API which will help injection?

Integration Diagram for Tellus and BrowserStack - Automated testing
Purple arrow show new features/implementations



[UPDATE] We no longer download test cases from Perforce ("P4") after migration to GitHub.

tellus Browserst...

XML for above diagram using https://www.draw.io/

Notes from meeting on Dec 1st

- Each single driver session at BS side will generate logs which can be viewed on BS Dashboard. - Logging
- Text Logs, Console Logs and Network Logs (which are basically in the form of text) are retained for 90 days on BS servers. Logging
- Screenshots and Videos will be stored for 30 days but you can get it later via API call and store them locally. Logging
- Video limit Logging
  - Can store long videos too but would be nice to have shorter videos.
- Will each test case have separate logs when running on one driver instance?
  - No, logs will be based on each driver session. Recommendation is to run one test case per driver instance
- What is the criteria to divide the no. of test cases to different browser instance?
  - BS does not control this, we can control this via script and specify how many scripts we want to divide per driver instance?
- **If we go by each driver instance for one test case how we'll be able to cope up with the limitation on concurrent BS instances which are 30 as of now??**
- **Each test case per instance would need many serial numbers which can exhaust our SN pool. What is the plan to handle this? Even if we are ready to allocate SN, is our infrastructure ready to serve the request that sooner?**
- **How we'll consolidate a number of result logs if we go by each test case per instance??**
- BS will hard stop job after 90 mins. And timeout after 90 secs if no command is received within that window. - Job Config
- It is possible to get BS logging Dashboard URL which can be used to launch the dashboard directly. - Reporting
- Session IDs can also be relied on to launch BS logging dashboard. - Reporting
- Upload and Download are not possible on BS servers still. - Upload and Download
  - BS launch a VM for each driver instance. They have 1000s of VMs running.

Mobile Testing

- Using API we can upload .apk files to BS server which will return you URL with hash key pointing to the uploaded file.
- Files uploaded will be stored forever and you can use it for multiple test instances.
  You can also store multiple versions of .apk files.

Joseph's Note:

- Can use FactSet corporate SSL CA root certificate
- **LIMA: Cannot run Fiddler in BS => Lilun is testing local testing w/browsermod proxy**
- Use localtesting.exe for local testing
- Can set cookies **Fiddler Injection and cookies**
- BS records video of session -- can handle long videos, but easier to find things w/shorter ones - Logging
- BS console text log is element level (but Tellus log is test case or assertion level) Logging
  - Can also access BS's log by REST URL (but URL has long hash key)
- **VM hard stop at 90 minutes** Logging
  - => Recommend parallel testing
- Console log, network logs retains 90 days. Text logs 30 days. Logging
- Only one browser runs in a VM. It is VM, not Docker container. Infra
- Need license key per manual UI access session (up to team license limit) - Licensing
- Infrequent maintenance downtime (twice a year, early Sunday morning), has fallback servers
- **Cannot export to Excel** (download) on BrowserStack machine -- no access to its filesystem Upload and Download

Mobile app

- we upload app binaries to BS, and get bs:<hashkey> URL, put that in caps['app'] = "bs:xxxxx"
  - API to list, update, delete apps
  - **Retention, size limit???**

Lilun's Questions:

- Can we configure 90 minutes' timeout? IE is working slow. Sometimes it may need more time to respond.
- Can multiple people access the dashboard using the same credential at the same time to check different jobs/logs?

Notes:

- Start the local server "**BrowserStackLocal.exe**" to do local testing. I tested this before.
- Play video for test cases' run. Can see what happens at each step.
- Capabilities are set in the script and have more browserstack-specific attributes.
  - Example for **local testing**:
    caps = Selenium::WebDriver::Remote::Capabilities.chrome
    caps['browser'] = 'Chrome'
    caps['browser_version'] = '59.0'
    caps['os'] = 'Windows'
    caps['os_version'] = '7'
    caps['resolution'] = '1920x1080'
    caps['acceptSslCerts'] = 'true'
    caps["browserstack.debug"] = "true"
    caps['browserstack.local'] = 'true'
- Hub url looks like <http://username:accesskey@hub-cloud.browserstack.com/wd/hub>
  - wd = Watir::Browser.new(:remote, url: "http://username:accesskey@hub-cloud.browserstack.com/wd/hub", :desired_capabilities => caps)
- Set cookies or Fiddler Header injection need to be looked into for resolution. BS has no experiences with that.
- Not sure whether BS can set trusted sites in IE.

BrowserStack Pricing and Other information

Monday, December 11, 2017 1:20 PM

Contact from Browserstack :
Matt Adams matt@browserstack.com

Pricing discussion with BrowserStack was done based on below assumptions

Test cases eligible to run on Selenium grid or Browserstack infra*	Current(Year 2017) # of Automated Test cases	Future (End 2019) # of Automated Test cases
Web (Desk Top)	2000	4000
Web (Mobile)	1000	2000
Mobile (Native app)	100	700
Browser (Desktop and Mobile)	Chrome	2
	IE	2
	FF	2
	Safari	2
Mobile (Native)	Android	1
	Apple	1

*QA Apps which are purely web apps and do not have any dependencies for initiation like running workstation in background etc need to be considered for Browserstack

Above are projection for # Of test cases run today and would be run in future.

Assumption: Each QA Test cases Average run time 8 mins to be finished in 48 hours

Definition of Parallel sessions:. *A "browser" is a combination of operating system (e.g., Windows), browser (e.g., Chrome) and version (e.g., 58). Any time any one of those three changes, it's another "browser" in the calculator for pricing*

Initial Pricing discussion started with 30 parallel extending to 60 parallel over 2 - 3 years

To: Rajul Shrivastava <rajul.shrivastava@factset.com>
Subject: Re: Updated Invitation: Matt and Rajul connect @ Fri Jul 7, 2017 8am - 8:30am (PDT) (rajul.shrivastava@factset.com)

Below are the quotes you requested. As I mentioned, the security review will assist in determining whether Enterprise or Pro would be the ideal solution.

Automate Mobile Pro		
Parallels	Monthy USD	Annual USD
30	$5,970	$71,640
61	$12,139	$145,688

Automate Mobile Enterprise		
Parallels	Monthly USD	Annual USD
30	$8,970	$107,640
61	$18,239	$218,868

Thanks,
Matt

----------Conversation for 25 and 28 Parallels----------

Overtime when discussed and presented to Jim arena, keeping in mind we wanted to start small and also check prices for 25 parallel and 28 parallel, below is the cost for same.
Below cost is for Automat Mobile and Automation Mobile Enterprise.

Hi Matt

Thanks for the invoices

Since I need all Desktop, Mobile Browsers and Native application testing BrowserStack platform I should be looking at below Automate Mobile pro and Automation Mobile enterprise correct ?

Automate Mobile:
25 Parallels: $59,988

28 Parallels: $66,864

Automate Mobile Enterprise:
25 Parallels: $89,700
28 Parallels: $100,464

Below are features mentioned by Matt Adams of Browserstack for Mobile Pro and Enterprise. ***Day 1 we should go with Mobile Pro as we get all we need with it.***

Also ***Automation Mobile pro comes*** with below features along with automation on real devices

- Unlimited Live
- Unlimited Screenshots
- Unlimited Responsive testing
- Unlimited Screenshots via API
- Unlimited Automate Desktop
- Unlimited Automate **Mobile***

Mobile testing features

- Unlimited app testing minutes
- Real Android and iOS Devices
- Natural gestures and interactions
- Range of iOS and Android versions
- Access to App Device Logs
- Smart app management
- Local testing

Automate Mobile Enterprise with below features along with automation on real devices

- Enterprise-grade security
- SSO integrations
- Advanced team management
- Usage reporting
- IP whitelisting
- Priority support
- Technical account manager ++ Plus other features as listed on your website

Mobile testing features

- Everything in App Live
- Enterprise-grade security
- SSO integrations
- Advanced team management
- Usage reporting
- IP whitelisting
- Priority support
- Technical account manager

Other less useful options could be

Automate Pro:
25 Parallels: $29,998
28 Parallels: $33,265

Automate Pro Enterprise:
25 Parallels: $44,700
28 Parallels: $50,064

Discussion around Mobile Native App Automated Testing

On Wed, Aug 30, 2017 at 3:25 PM, Matt Adams <matt@browserstack.com> wrote:
Hi Rajul,

I would need 2 parallel users for Automated testing of Native app, 1 for Android device and 1 for apple.
App Live: 2 Parallels = $1788 - Recommended
App Live Enterprise = $4499

When would App Automate (Automated testing ability for Native app) available?
App Automate is should be released by the end of the year (our PM's won't commit to a hard release date). However, they did open the Alpha up to paying customers, so we would be able to get you access prior to GA.

Also Automate Mobile Pro and Enterprise provide simulator support for mobile testing ?
No, we are deprecating our simulators in favor of all real devices. This is being done for 3 reasons:

Performance: Tests run on physical mobile devices are significantly faster, often up to 50% faster compared to emulators and simulators, resulting in lower build and test times for our customer.

Stability: Emulators and Simulators typically generate a lot more errors when compared to real devices. By testing on real devices, users will experience higher stability, resulting in fewer errored tests.

Accuracy: Real devices are the most accurate reflection of the user experience. Emulators and simulators often give false positives and false negatives, resuting in bugs making it to production, and time lost debugging "false" bugs.

Thanks,
Matt

Email conversations

Monday, December 11, 2017 2:00 PM

NDA was in Place with BrowserStack

NDA_Brows
erStack_F...

Communication on pricing for Native app testing and automated testing listed below in email.

Re
Updated I...

Email on Native app Automated Testing availability

BrowserSta
ck POC
k

Documentation

Here is the documentation for the topics we discussed on the call:

- https://docs.google.com/a/browserstack.com/forms/d/11lIJHrJ6h7dz4Ku0_g1cdJK4D7vF39ykYXBp-WwSm7w/viewform?edit_requested=true
- Downloading files during Automate
- Local Testing Modifiers
- Selenium Capabilities
- Browsers and Devices for Automate

Local Testing documentation

Re
BrowserS...

Draft: Apps decided for POC

Tuesday, December 19, 2017 7:41 PM

App Eligible
to run on ...

Applications shortlisted for POC. Criteria: Test Case count is b/w 10-25

Portfolio	S.no	Name of Application	# of Total Test cases	Eligible to run on Browserstack/Selenium Grid	POC Candidate	No Lima Overrides Required? (Yes/No)
Analytics	3	PDB2	25	Yes	Yes	
Core	15	Ownership	24	Yes	Yes	
Core	17	Filings	12	Yes	Yes	
Core	21	Industry	14	Yes	Yes	
Core	25	Call Street	13	Yes	Yes	
Core	26	Idea Screening	16	Yes	Yes	
Platform	30	OA3(Online Assistant)	19	Yes	Yes	
Platform	35	Formula lookup	11	Yes	Yes	

Discussion with Tellus Team

Friday, January 05, 2018 6:02 PM

- For POC, Tellus will enable scheduling jobs on BrowserStack via its interface.
- After POC, for full fledge execution we need more parallel session but we are having a contract of only 30 sessions. Thus, queuing is required which is not stable on tellus side.
    - NOTE: BS itself does not provide any queuing mechanism.
- What if more than available resources are requested?
    1. Should we drop excessive requests?
    2. Add them to queue and invest on stabilizing queuing mechanism?

Documentation

Friday, January 05, 2018 7:47 PM

Configuration changes needed for Protractor Projects:
https://www.browserstack.com/automate/protractor

BrowserStack Local Testing

Tuesday, January 09, 2018 8:42 PM

In order to test scripts on Browser Stack one has to start **BrowserStackLocal.exe** as follows:
BrowserStackLocal.exe --key <access_key> --force-local --local-identifier <identifier_name (any name can be given)>

Notes:

1. If you do not use **--force-local** option you'll see error on BS browser instance saying "You cannot login from your current IP address" while logging into FactSet page.
2. **--local-identifier** flag is needed if you want to run another application using the same account.
3. You need not run **BrowserStackLocal.exe** on each machine. Say, you started **BrowserStackLocal.exe** with your credentials on your machine (along with **--local-identifier** option), you can ask your colleague to use your credentials (which is BrowserStack's **username** and **accesskey** here. You can find this information after login into your account at **browserstack.com** under **Settings** section) to run the application from his machine (without launching **BrowserStackLocal.exe** on his machine). However, you need to make the changes in configuration of your project which will be described below.
4. With trial account, I observed that when I submitted more jobs than allotted limit, it is able to queue the remaining jobs. **I had 5 parallel sessions and 5 jobs were allowed to queue. But, when queue gets filled the next jobs will terminate immediately.**

Configuration Changes Needed
For Ruby:
Add the following configuration to **Config.rb**

```
webCaps['os'] = 'Windows'
webCaps['os_version'] = '7'
webCaps['resolution'] = '1920x1080'
webCaps['acceptSslCerts'] = 'true'
webCaps["browserstack.debug"] = "true"
webCaps['browserstack.local'] = 'true'
webCaps['browserstack.localIdentifier'] = 'Test123'
```

Hub URL format mentioned in Ruby:
http://browserstack_username:browserstack_accesskey@hub-cloud.browserstack.com/wd/hub

For Protractor:
Add following Key-Value pairs to your configuration file

```
exports.config = {
  'seleniumAddress': 'http://hub-cloud.browserstack.com/wd/hub',
'capabilities': {
   'browserstack.user': '<browserstack_username>',
   'browserstack.key': '<browserstack_accesskey>',
   'browserstack.local': true,
   'browserstack.localIdentifier' : '<local_identifier_name>',
   'browserstack.debug':true,     //optional
   'os': 'Windows',
   'os_version': '7',
   'browserName': 'Chrome',
   'browser_version': '63.0',
```

```
  'resolution': '1920x1080',    // optional
  'chromeOptions': {
       'args': ["--disable-plugins", "--start-maximized"]    // optional
   },
 }
};
```

Note: If you don't use **resolution and chromeOptions** in the configuration, browser will still start maximized.

Takeaway
Once **BrowserStackLocal.exe** is started on **bstackdeva01** server with service account (which will be created for Tellus usage), we'll be able to submit jobs from Tellus JEP. **This has to be tested yet.**

Question?
Does individual will have personal **BrowserStack** account to test script locally while development?

Local Testing

Tuesday, January 09, 2018 8:42 PM

In order to test scripts on Browser Stack one has to start **BrowserStackLocal.exe** as follows: **[DO NOT USE THIS ON YOUR MACHINE]**
BrowserStackLocal.exe --key <access_key> --force-local --local-identifier <identifier_name (any name can be given)>

Notes:

1. If you do not use **--force-local** option you'll see error on BS browser instance saying "You cannot login from your current IP address" while logging into FactSet page.
2. **--local-identifier** flag is needed if you want to run another application using the same account.
3. You need not run **BrowserStackLocal.exe** on each machine. Say, you started **BrowserStackLocal.exe** with your credentials on your machine (along with **--local-identifier** option), you can ask your colleague to use your credentials (which is BrowserStack's **username** and **accesskey** here. You can find this information after login into your account at **browserstack.com** under **Settings** section) to run the application from his machine (without launching **BrowserStackLocal.exe** on his machine). However, you need to make the changes in configuration of your project which will be described below.
4. With trial account, I observed that when I submitted more jobs than allotted limit, it is able to queue the remaining jobs. **I had 5 parallel sessions and 5 jobs were allowed to queue. But, when queue gets filled the next jobs will terminate immediately.**

Configuration Changes Needed
For Ruby:
Add the following configuration to **Config.rb**

```
webCaps['os'] = 'Windows'
webCaps['os_version'] = '7'
webCaps['resolution'] = '1920x1080'
webCaps['acceptSslCerts'] = 'true'
webCaps["browserstack.debug"] = "true"
webCaps['browserstack.local'] = 'true'
webCaps['browserstack.localIdentifier'] = 'Test123'
```

Hub URL format mentioned in Ruby:
http://browserstack_username:browserstack_accesskey@hub-cloud.browserstack.com/wd/hub

For Protractor:
Add following Key-Value pairs to your configuration file

```
exports.config = {
  'seleniumAddress': 'http://hub-cloud.browserstack.com/wd/hub',
'capabilities': {
   'browserstack.user': '<browserstack_username>',
   'browserstack.key': '<browserstack_accesskey>',
   'browserstack.local': true,
   'browserstack.localIdentifier' : '<local_identifier_name>',
   'browserstack.debug':true,     //optional
   'os': 'Windows',
   'os_version': '7',
   'browserName': 'Chrome',
```

```
  'browser_version': '63.0',
  'resolution': '1920x1080',    // optional
  'chromeOptions': {
       'args': ["--disable-plugins", "--start-maximized"]    // optional
   },
 }
};
```

Note: If you don't use **resolution and chromeOptions** in the configuration, browser will still start maximized.

Takeaway
Once **BrowserStackLocal.exe** is started on **bstackdeva01** server with service account (which will be created for Tellus usage), we'll be able to submit jobs from Tellus JEP. **This has to be tested yet.**

Question?
Does individual will have personal **BrowserStack** account to test script locally while development?

Test mobile apps with BrowserStack

Thursday, January 11, 2018 11:17 AM

Android apps

1. Upload the android apk file to BrowserStack using the following command:

   ```
   curl -u "browserstack_username:browserstack_accesskey" -X POST "https://api.browserstack.com/app-automate/upload" -F "file=@C:\browserstack\tests\android\WikipediaSample.apk"
   ```

   This command will return id for the app on browserstack like:

   ```
   {"app_url":"bs://172c0cf752e39a9728fc8b9a6140c968a85c53c9"}
   ```

2. Modify the script like this (in ruby):

   ```
   username = 'browserstack_username'
   access_key = 'browserstack_accesskey'

   caps = {}
   caps['build'] = 'Ruby Appium Sample'
   caps['name'] = 'single_test'
   caps['device'] = 'Google Nexus 9'
   caps['browserstack.debug'] = true # optional
   caps['app'] = 'bs://172c0cf752e39a9728fc8b9a6140c968a85c53c9'

   appium_driver = Appium::Driver.new({
        'caps' => caps,
        'appium_lib' => {
             :server_url => "http://#{username}:#{access_key}@hub-cloud.browserstack.com/wd/hub"
        }})
   ```

3. Sample script

   sample_sin
   gle_test

BrowserStack PoC - Pending Issues

Tuesday, January 23, 2018 2:01 PM

RPD:34302004 Browserstack Evaluation Firewall Needs
Need to whitelist URLs on bstackdeva01.

How To: BrowserStackLocal.exe with proxy server

Thursday, January 18, 2018 9:03 AM

For BrowserStackLocal.exe to use proxy server:

```
BrowserStackLocal.exe  --key GgvPzJsHnDgaVpozsH2E --local-proxy-host 127.0.0.1 --local-proxy-port 8888 --force-proxy
```

Reference

- https://www.browserstack.com/local-testing
- http://docs.telerik.com/fiddler/Configure-Fiddler/Tasks/UseFiddlerAsReverseProxy