Mozilla iris

A ui/ux proposal for the Mozilla Iris Control Center

(Outreachy18)

by Ioannis Valasakis

@wizofe <code@wizofe.uk>

Mozilla Iris - A Visual Automation Tool

DETAILS

Outreachy and me

Name: Ioannis Valasakis

IRC Handle: wizofe

Email: code@wizofe.uk

User Page: https://mozillians.org/en-US/u/wizofe/

Location: United Kingdom

Time Zone: GMT+1

Mentors: Matt Wobensmith <mwobensmith@mozilla.com>

RESEARCH & SYNOPSIS

The reality

It seems that the current design of Iris Control Center (landing page) isn't satisfying for the developers of Mozilla Iris Automation Tool. This results from different issues:

- The presentation of collected data is not intuitive and doesn't give an overview of the results
- The structure of the page isn't clean
- There is no inline validation on the forms
- Not apparent (if any) keyboard shortcuts that let the user move around the website or perform actions by only using the keyboard
- Not clearly defined user stories at the present state

Some of those problems have been discussed on the GitHub Issue #963 (and #453) and noted down here.

What follows, is a proposal that will attempt to fix, engage with the developers' team, iterate and understand deeper the above-described issues.

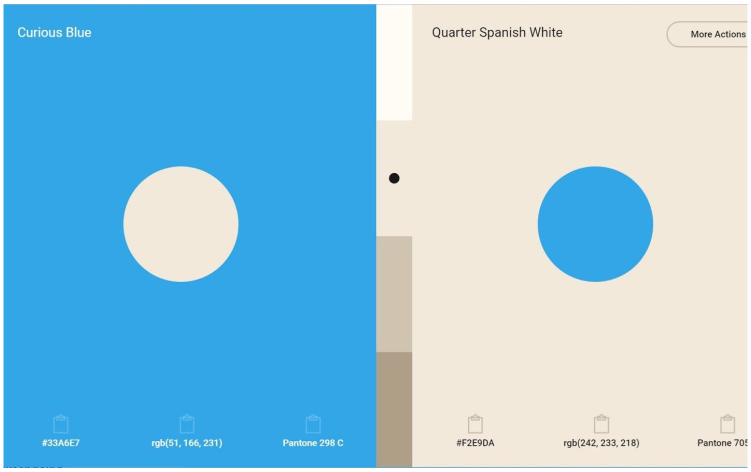
DESIGN

Talking handsome

The ideation phase was a brainstorming session as well an investigation.

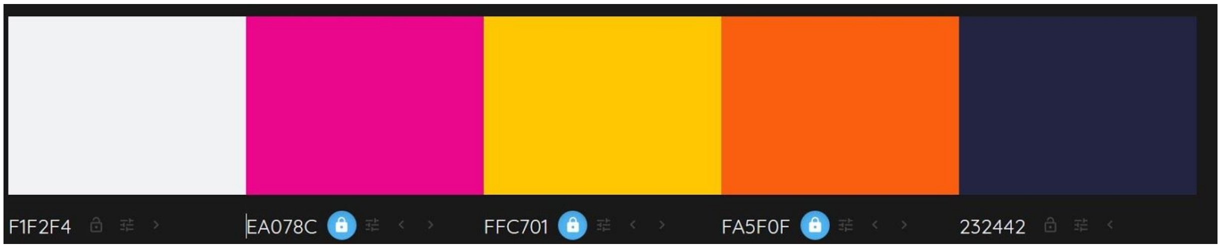
Software developer friends proposed their favourite automated GUI testing. I also looked at online on current GUI testing solutions.

Some of the attached images, are taken from real-life projects, but that's only as a part of the ideation process and because the proposal time is limited.



As a colour palette, I propose the combination of Curious Blue with Quartier Spanish White/Pure White (not shown), clean and direct approach. It is minimal, affirmative but also refreshing and brings attention to the details.

Below is another style option (or an extended one). This is a more vibrant option. It’s created by sampling the three primary colours of the new Mozilla Logo proposal and comlimenting them with a lighter (main text) and a darker option.

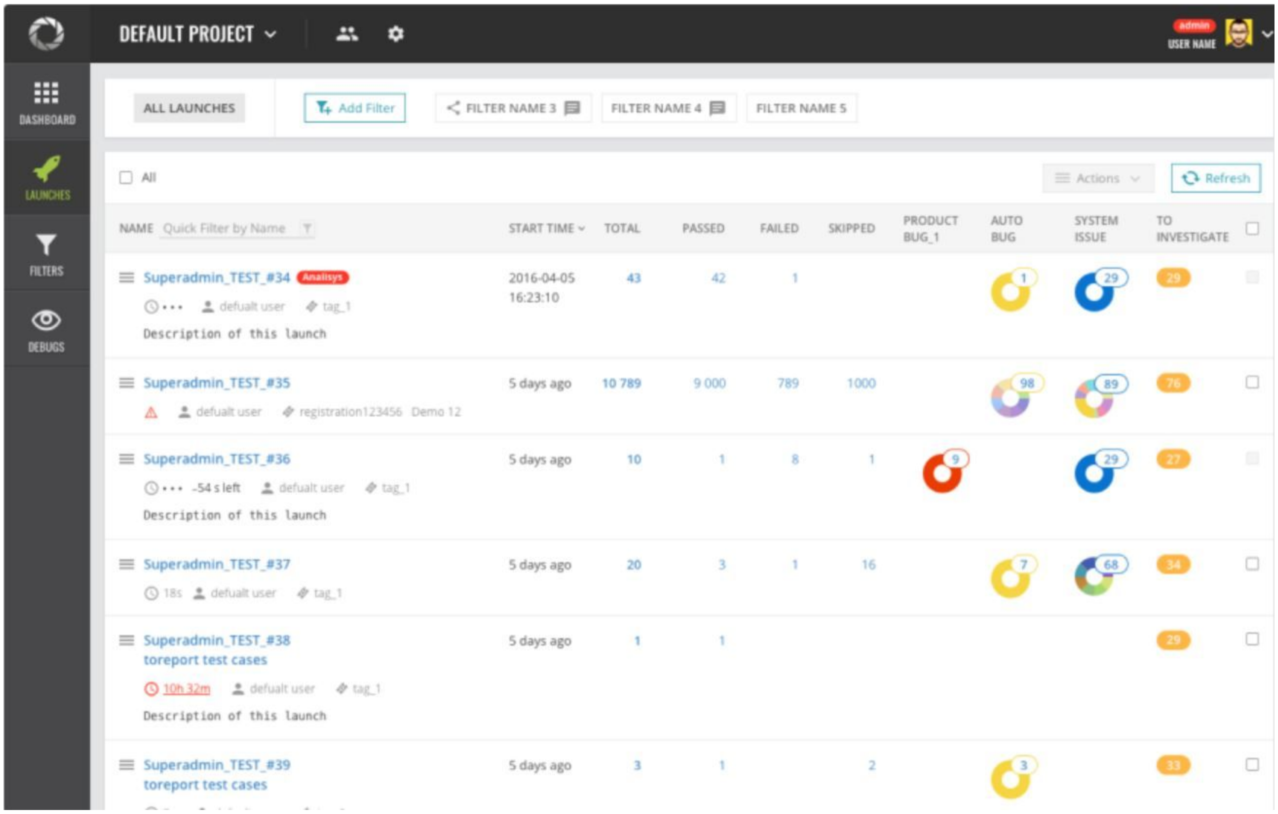


Main structure

A left-hand side menu, with access to the latest tests, test configuration, etc. would make the Iris Control Center a very user-friendly destination.

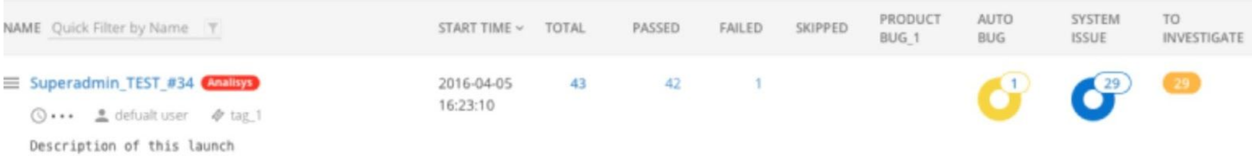
The centre of attention is the right frame, would be the tests, sorted by date and time (but that can be configured separately if needed). The user can see with one glance, how many times are passed or failed, with which configuration as well as physical pie diagrams with more specific test failure/success rates.

That is a very informative approach and on the same time an extremely different route from the one that’s currently followed.



NAME Quick Filter by Name	START TIME	TOTAL	PASSED	FAILED	SKIPPED	PRODUCT BUG_1	AUTO BUG	SYSTEM ISSUE	TO INVESTIGATE	☐
Superadmin_TEST_#34 Analisys ••• defualt user tag_1 Description of this launch	2016-04-05 16:23:10	43	42	1			1	29	29	☐
Superadmin_TEST_#35 ⚠ defualt user registration123456 Demo 12	5 days ago	10 789	9 000	789	1000		98	89	76	☐
Superadmin_TEST_#36 ••• -54 s left defualt user tag_1 Description of this launch	5 days ago	10	1	8	1	9		29	27	☐
Superadmin_TEST_#37 18s defualt user tag_1	5 days ago	20	3	1	16		7	68	34	☐
Superadmin_TEST_#38 toreport test cases 10h 32m defualt user tag_1 Description of this launch	5 days ago	1	1						29	☐
Superadmin_TEST_#39 toreport test cases	5 days ago	3	1		2		3		33	☐

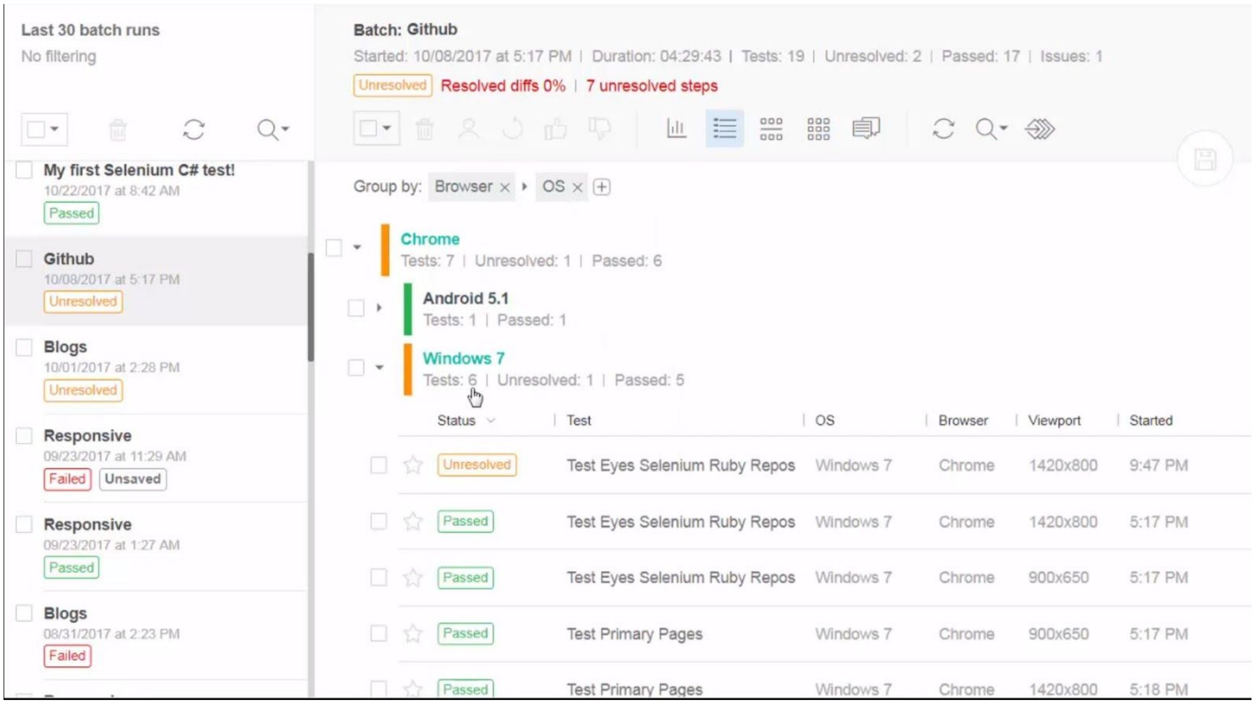
Here is a zoom in, on the main panel showing the details of the tests:



NAME Quick Filter by Name	START TIME	TOTAL	PASSED	FAILED	SKIPPED	PRODUCT BUG_1	AUTO BUG	SYSTEM ISSUE	TO INVESTIGATE
Superadmin_TEST_#34 Analisys ••• defualt user tag_1 Description of this launch	2016-04-05 16:23:10	43	42	1			1	29	29

It is also possible if more information is needed to create a more layered structure/tree whereby clicking the test performed gives a new website with even more details about it.

An alternative presentation, where there’s no configuration and left menu but a framed window, where the menu is replaced by the tests performed (sorted again by date/time). The main frame on the right side, it gives information in much detail, in a tree-like format (quite similar to the current one but much cleaner. Using coloured labels helps to identify failed tests or tests that need a resolution quickly.



Last 30 batch runs
No filtering

□ My first Selenium C# test!
10/22/2017 at 8:42 AM
Passed

□ Github
10/08/2017 at 5:17 PM
Unresolved

□ Blogs
10/01/2017 at 2:28 PM
Unresolved

□ Responsive
09/23/2017 at 11:29 AM
Failed Unsaved

□ Responsive
09/23/2017 at 1:27 AM
Passed

□ Blogs
08/31/2017 at 2:23 PM
Failed

Batch: Github
Started: 10/08/2017 at 5:17 PM | Duration: 04:29:43 | Tests: 19 | Unresolved: 2 | Passed: 17 | Issues: 1
Unresolved Resolved diffs 0% | 7 unresolved steps

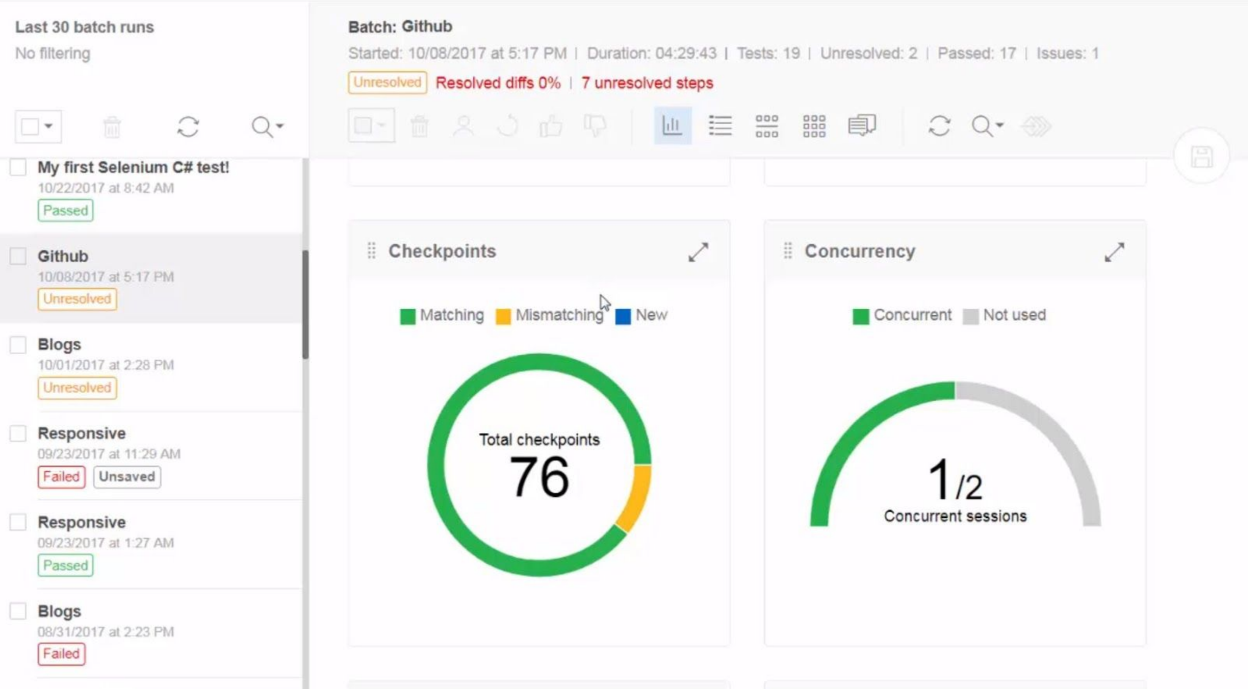
Group by: Browser × ▸ OS × ⊞

□ ▾ Chrome
Tests: 7 | Unresolved: 1 | Passed: 6

□ ▸ Android 5.1
Tests: 1 | Passed: 1

□ ▾ Windows 7
Tests: 6 | Unresolved: 1 | Passed: 5

		Status	Test	OS	Browser	Viewport	Started
□	☆	Unresolved	Test Eyes Selenium Ruby Repos	Windows 7	Chrome	1420x800	9:47 PM
□	☆	Passed	Test Eyes Selenium Ruby Repos	Windows 7	Chrome	1420x800	5:17 PM
□	☆	Passed	Test Eyes Selenium Ruby Repos	Windows 7	Chrome	900x650	5:17 PM
□	☆	Passed	Test Primary Pages	Windows 7	Chrome	900x650	5:17 PM
□	☆	Passed	Test Primary Pages	Windows 7	Chrome	1420x800	5:18 PM



Last 30 batch runs
No filtering

My first Selenium C# test!
10/22/2017 at 8:42 AM
Passed

Github
10/08/2017 at 5:17 PM
Unresolved

Blogs
10/01/2017 at 2:28 PM
Unresolved

Responsive
09/23/2017 at 11:29 AM
Failed Unsaved

Responsive
09/23/2017 at 1:27 AM
Passed

Blogs
08/31/2017 at 2:23 PM
Failed

Batch: Github
Started: 10/08/2017 at 5:17 PM | Duration: 04:29:43 | Tests: 19 | Unresolved: 2 | Passed: 17 | Issues: 1
Unresolved Resolved diffs 0% | 7 unresolved steps

Use cases

Use cases may include the following:

- Parallel testing: Testing different versions of Firefox (release, beta, nightly) and on different platforms
- Agile testing: Ensure that small changes don’t break the UI
- Continuous testing: Iris can be a part of a CI pipeline
- Automated regression testing: The main focus of Iris at the moment. Automating tedious and very manual testing from Testrail.

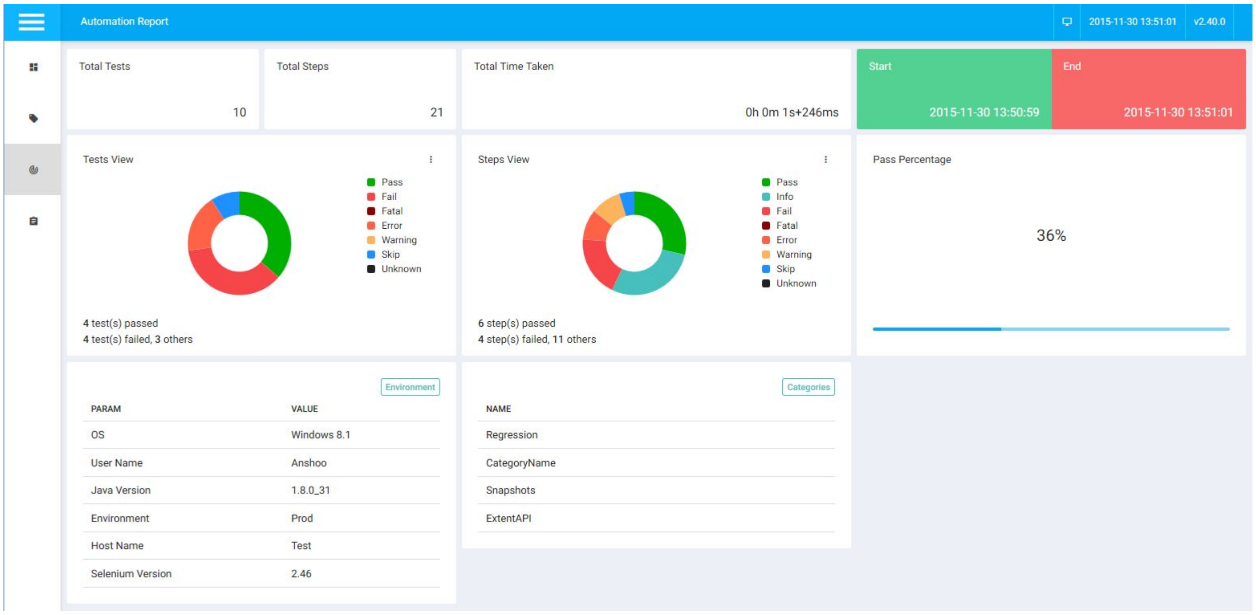
During the term of the Outreachy Internship, I want to define and explore even more test cases with the help of Iris developers’ and the community.

Data presentation

All the tests results are in a hidden folder as JSON files under the iris/data folder. Their representation is a very simple JSON parsing and table rendering. That doesn't offer a good experience for the end user. How can this be refined?

A starting point is the UI described above, with a left pane with settings about the run tests. I'll show here two possible approaches:

A – THE PIE DIAGRAM



Total Tests 10

Total Steps 21

Total Time Taken 0h 0m 1s+246ms

Start 2015-11-30 13:50:59

End 2015-11-30 13:51:01

4 test(s) passed
4 test(s) failed, 3 others

6 step(s) passed
4 step(s) failed, 11 others

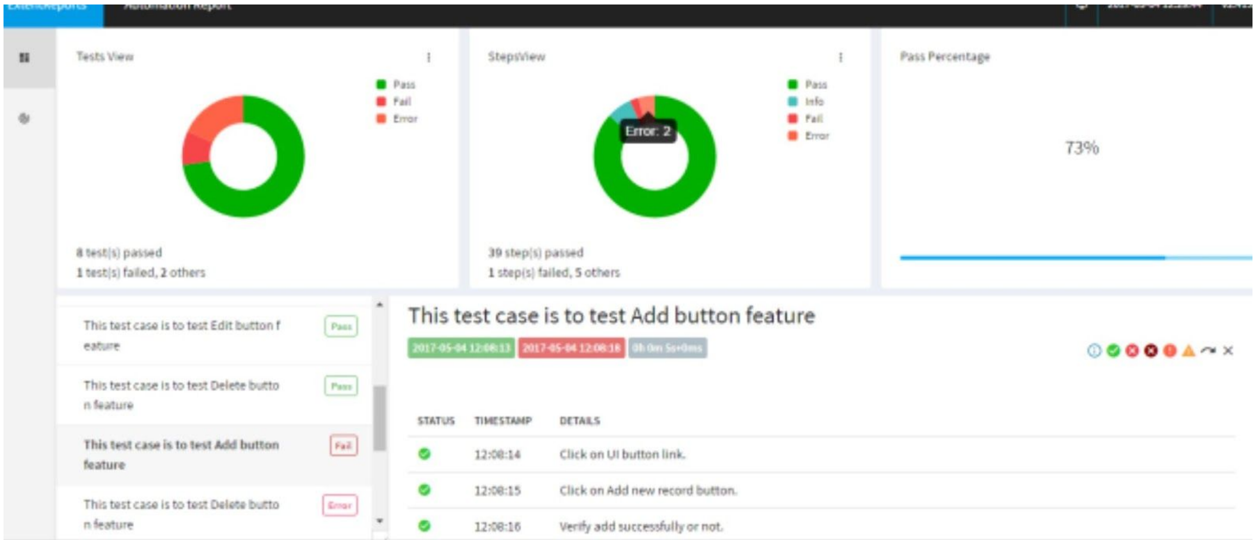
Pass Percentage 36%

PARAM	VALUE
OS	Windows 8.1
User Name	Anshoo
Java Version	1.8.0_31
Environment	Prod
Host Name	Test
Selenium Version	2.46

NAME
Regression
CategoryName
Snapshots
ExtentAPI

Here, a summary of the results is displayed via a colourful pie diagram. By clicking the different colours of the pie, a new view opens and describes in detail the relevant test.

The image below visually describes that:



8 test(s) passed
1 test(s) failed, 2 others

39 step(s) passed
1 step(s) failed, 5 others

This test case is to test Edit button feature Pass

This test case is to test Delete button feature Pass

This test case is to test Add button feature Fail

This test case is to test Delete button feature Error

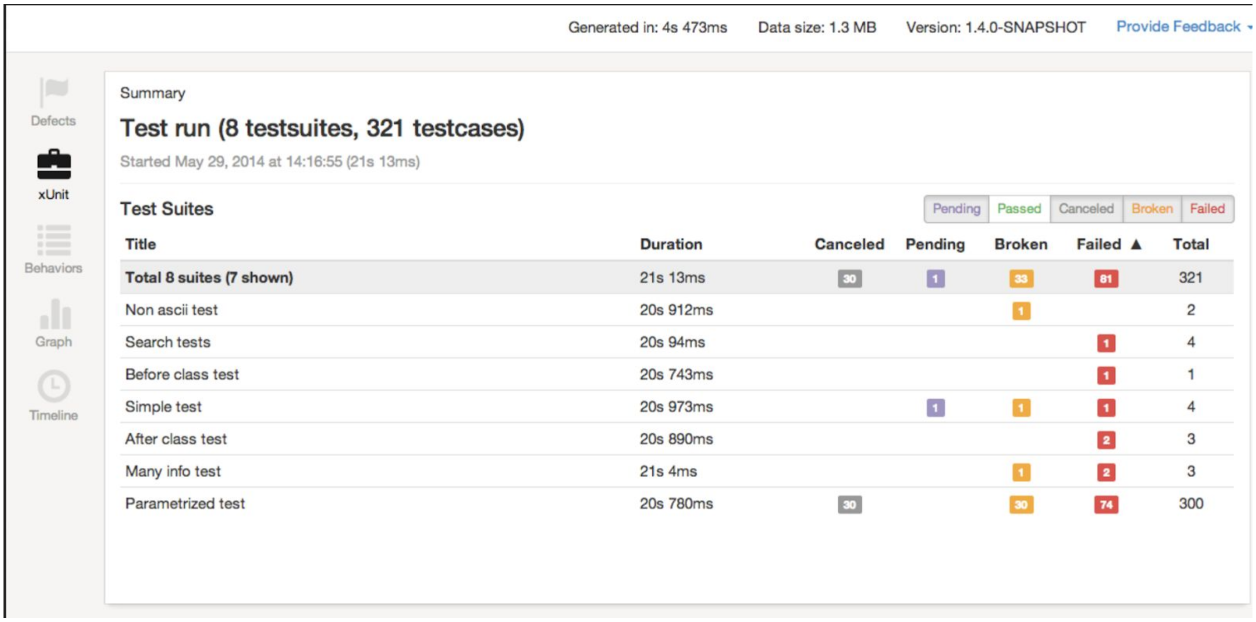
This test case is to test Add button feature

2017-05-04 12:08:13 2017-05-04 12:08:18 0h 0m 5s+0ms

STATUS	TIMESTAMP	DETAILS
✓	12:08:14	Click on UI button link.
✓	12:08:15	Click on Add new record button.
✓	12:08:16	Verify add successfully or not.

B – THE LABELLED PANE

This option displays a more minimal result, oriented in faster user interaction. The description of the test status is done via coloured labels. Test results are sorted and by status duration and title descriptors. Additionally, a submenu with the descriptors is shown directly above the tests, to see in the details of those tests.



Generated in: 4s 473ms Data size: 1.3 MB Version: 1.4.0-SNAPSHOT

Summary

Test run (8 testsuites, 321 testcases)

Started May 29, 2014 at 14:16:55 (21s 13ms)

Test Suites

Title	Duration	Canceled	Pending	Broken	Failed ▲	Total
Total 8 suites (7 shown)	21s 13ms	30	1	33	81	321
Non ascii test	20s 912ms			1		2
Search tests	20s 94ms				1	4
Before class test	20s 743ms				1	1
Simple test	20s 973ms		1	1	1	4
After class test	20s 890ms				2	3
Many info test	21s 4ms			1	2	3
Parametrized test	20s 780ms	30		30	74	300

PLANS

Looking at the future

Discovery Phase: currently

Definition Phase: 1-2 weeks

Design Phase: 4 weeks

Implementation Phase: the rest of the project

Testing: 3-4 weeks

DELIVERABLES and OUTCOMES

Giving to the community

- Gain a deeper understanding of Iris project and in particular the Iris Control Center and its interface
- Get an understanding of how the project and the codebase is structured and how primary test cases are implemented
- Learn more about the inside workings of Mozilla Firefox, React and Iris
- Formulate a coherent design that tackles current issues and improves the design and its usability
- Create mock-ups, potential new interfaces and workflow
- Iterate with the Iris team to define precisely the aims
- Work with the Iris team to see those ideas implemented in code, providing feedback and testing

FEW WORDS

About my experience

I started getting involved with computers since I was 9 and my mom had at work an IBM PC running DOS and some ncurses like librarian's software. I got hooked since the first moment I saw a cursor blinking on the screen. The unknown, the beauty of the electron beams on a CRT screen and displaying typed characters on that monitor, made me fall in love with computing.

Since that time I studied Electronics Engineering and did an internship in an R&D company. I never finished that degree though due to personal reasons that at the time didn't help me go through. Many things and changes happened between then and now. Ten years forward and I've decided to come back to studying. Currently, I am completing my MSc in Bioinformatics at the University of London, and I'm proud of this achievement. It came through hard work and part-time work, as I had to support myself financially. I learned a lot and overcame struggles (I was also fortunate to be able to get some counselling through my University when I needed it).

I've been involved with Open Source since the 1st Open Source/Free Software Conference in Greece, where we invited Richard Stallman! That was my first real experience with the social aspect of things (real-life social like meeting other amazing developers). My first contribution was then translating the GNU GPL in Greek!)

My specialisation lies in Embedded Software, Linux, C/C++ and Python (oh, I should state, since I am applying to Mozilla, my loooove of Rust but I have still a long way to go!). I also have some experience with Jenkins and briefly involved with AWS S3 buckets by writing a script to back-up and deploy images build for an Embedded Open Source OS based on the Raspberry Pi.

Involved briefly with Debian (and its packaging system!). I tried to host my own package; still, haven't found a mentor to support it though; but I am lintian clean (v0.8-1) which is a pride
https://mentors.debian.net/package/fsdiff

I also worked last summer for Redox (www.redox-os.org) on an internship to examine if an aarch64 version can be ported on silicon (specifically on the HiKey960 reference platform).

A few more about this, you can read on my blog post here:

https://medium.com/@wizofe/rsoc-porting-redox-to-arm-aarch64-0x02-c566ee22f377

I hope that you'll give me the chance to create something amazing together, for Mozilla and its vibrant community! I am thrilled by the fact that Mozilla fights for a secure and democratised Internet while being so awesome and at the same time, by creating high-quality software! It is my dream to be able to work for Mozilla :) :)

Hope to hear from you soon!

Regards,

Ioannis