Interview report

1. Choice of frameworks:

I have choosed to work with a **custom-made framework** because it offers a much greater control over most of the off-the-shef solutions that can be found. It is also free, using only open-source libraries and it is very customizable, expandable and maintainable at the expense of time to develop, learning curve, documentation.

The language choice was made purely from a convenience standpoint. I am most familiar with Java and I already know my way around most of the libraries used in this pack. Java provides plenty of resources on-line, easy to find and understand. Like thousands of free libraries, a vast community of professionals that can answer questions that might occur, or already faced some of the issues that may occur and already have a solution.

I've made the process of testing as seamless as possible using a CI/CD pipeline in **Jenkins** for one particular reason, and this is generally my mind-set as a SDET. Not all QAs know how to program. They might know how to read code and how to alter some of that code, but the job of a QA is to be creative. Not to spend hours developing a new script, or semi-manually run automated tests. All tests are just a click away, and they run on dedicated machines somewhere in the cloud. They now have free time to develop other test-cases and they have the resources of the local machine free to do other stuff. We all have our small collection of Chrome tabs opened and tens of applications already open and starved for resources in our machines.

Deploying everything in a single machine is not always the best solution. Every machine is different and might crash or have some piece of software installed that causes issues. That's why **Docker** (or any other containerization solution) seemd fitting for this kind of project. I chose to deploy everything in containers. The application in a dedicated container, the Selenium Grid in other containers (hub and nodes separate), the Jenkins instance in another one. This way if something goes wrong, it's just a few clicks away or two scripts to get everything up and running again with full control over the environment.

Running a single test is a piece of cake for any machine. Even a mildly complex one like this. And maybe you wander why **Selenium Grid**?. Well, not only the framework needs to be robust and expandable. The test infrastructure is at least as important. Today we might have only one test-case. Tomorrow will have 10 scripts. They will run serially (one after the other) with no issues in a matter of minutes. But in a year from now… we might have thousands of front-end scripts running. And a few minutes adds app. Fast! Before you know it, a Full Regression suite takes days. Maybe a week. (ask me how I know and how I fixed that :)). By having a robust infrastructure, a Grid of nodes on powerful machines that is one click away from running a suite of tests, we can save alot of time. All of the tests are fully independent of each other because remember, we have full control over the framework and other tools. All we have to do is compile a report at the end of the execution and display it in a stakeholder-friendly manner. That's it!

2. Instructions:

All of the instructions needed can be found inside the repository here: https://github.com/RaduSimonica/NetDataInterview/blob/master/README.md

3. Smoke test plan:

I didn't have enough time to fully explore and understand the netdata application. But from what I read on the internet about the application I might have an ideea of what needs to be done.

For the next release I would focus on the newly developed features, running the "happy path" tests on all of them. But first of all I'll run a security check on the full app. Then I'll include a few regression tests where the new features integrates with already existing code. After that, I'll run a few tests on the most used parts of the application. Hopefully the Analytics team can help, providing us a brief report on what the majority of our users is doing in our app.

If any new feature or changes affects the compatibility of the app with certain environments, I will include a few tests in that area as well.

But most importantly, due to the limited time for testing, immediately after the release, I will focus on a Full Regression suite for any possible issue that needs to be addressed asap. And keep an eye on the community forums. The users are the best testers after all. They always find issues.

4. Possible improvements.

Oh, that's quite a subject :). I have many improvements to do in the solution developed here. But I'll try to be brief.

- Add a reporting module;
- Add compatibility with all major operating systems;
- Add logging on all areas of the framework;
- Add exception handling on all methods;
- Add unit-tests on most methods and classes;
- Add functional tests on most methods and classes;
- Refactor the "BaseClass" to be more robust;
- Move the Alert creation inside the @BeforeSuite method (and add it).;
- Improve the Alert creation module with a bit of randomization;
- Handle failed tests (add retries for example);
- Improve the POM. Everytime a link is clicked, instantiate the expected object and return it;
- Add retries on click methods;
- Improve the wait methods;
- Add parameters to the Jenkins pipeline (so you don't need to manually add the local IP in the config file);
- Add Data Providers and refactor the tests to use them;
- Handle the logs and test results in a database. Log analytics about the executions (timestamps, failure reasons, exceptions, data used for testing, etc)

That's it for now. I can think of many more improvements, but I think that will suffice for now.

Thank you for the opportunity. It's been really fun for me (even though frustrating due to my local environment issues). I've learned a lot during this test. Both about the application and about the tools. Docker is one very good example. I cannot use Docker where I work right now and I've tried to learn it at home, by myself.

As a note: You have an awesome community around the application. And a top-notch documentation for the app and the API. I was able to find anything I needed on Google. That's great!

Hope to hear from you soon,
Radu Simonică

- Don't hate, Automate! -