Automated Testing Framework for Enketo

Patrick McCardle, Matendo Rugema, and Kyle Sheeley
Department of Computer Science, College of Charleston, Charleston, SC 29401

Introduction to Enketo

Enketo Express is a prime component in data collection, data storage, and data transfer. Enketo has been used all around the world for a wide range of uses from needs assessments in humanitarian aid, to raising historical awareness, to clinical research, and election monitoring. Enketo can be combined with other tools to flexibly create a full-fledged information management system. This has resulted in the adoption of Enketo into many systems. It also means that Enketo can focus on just data collection and do this collection in the best way possible.

Choosing Enketo

Enketo was not our first choice when it came to open source projects that we could build an automatic testing framework for. First, we chose to look at the Sahana Eden project. We ran into issues with the server and client platform that the Sahana Eden project used. One example is that to view test results using their platform was impossible because it required administrator rights on their server, which we could not get. After this failure, we looked into Kobotoolbox, which is an open source platform that works on data managment, and this is where we found Enketo Express.

The Process

The installation of Enketo Express is not that difficult. Although, as the installation can be quite long if described the whole way through, we will not cover it.

Instead, we will focus on the components that we used to make the automatic testing framework and those components of Enketo Express that hold importance to us.

First is Docker and Docker Compose. Docker is way for individual projects to be contained within a container. A container will contain everything that is required for the project to run at maximum performance. Its purpose is to ease the process by which the requirements and dependencies are managed. Docker Compose works with Docker to define and run the containers that hold all the requirements and dependencies necessary to run the project. Together they allow projects to be more easily managed and tested, which is why we used them in our project.

The other project components that help with testing are Grunt, Mocha, and Chai. Grunt helps prepare assets for testing, which helps reduce the time it takes to produce test results. Mocha allows for tests to be run asynchronously and makes the compilation of test results easier, which helps when running tests in rapid and often succession. Chai is an assertion library which allows for testing to be more adept.

Building an Automated Testing Framework

The design of Enketo automated framework was completed and all necessary 25 test cases were generated.

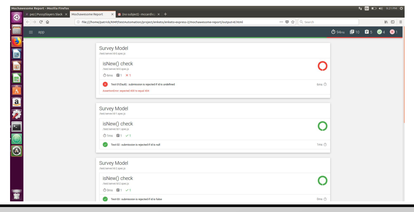
Methods Tested:
isNew()
set()
get()
get_cached_instance()

After generating all 25 tests from the above methods, we were able to get the Pass result on all the methods.

Software Fault Injection Testing

As for the Fault Injection, our team managed to introduce faults into the code to test the system robustness of error handling capabilities. This is to ensure that the system can be capable of handling and recovering from fault or error conditions and identifies any weaknesses.

In the example below, our test cases failed when faults were introduced into the code.



Conclusion

Developing an automatic framework for the Enketo Express project was exceedingly more difficult than we thought it would be. Many man hours of thinking and rethinking solutions and structures eventually resulted into a functional testing framework.

Works cited

https://www.docker.com/what-container
https://docs.docker.com/compose/overview
https://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/
https://www.humanitarianresponse.info/en/applications/kobotoolbox
https://gruntjs.com/
https://mochajs.org/

Acknowledgments

Dr. Bowring as our Main Supervisor for this project.

Further information

Any additional questions should be addressed to Kyle Sheeley at sheeleykw@g.cofc.edu.