

Automation Framework for Pub/Sub Validation

Sharankumar

Dept. of Computer Science and Engineering

RV college of engineering

Bengaluru, India

Wuppuluri, Arun Kumar

Senior Manager Engineer

Zebra Technologies

Bengaluru, India

Shankar k, Vinod

Software Engineer II

Zebra Technologies

Bengaluru, India

Dr Mamatha T

Assistant Professor

Dept. of Computer Science and Engineering

RV college of engineering

Bengaluru, India

Abstract – This study aims to find out a more efficient way of performing Pub/Sub Validation. Automation helps us to reducing manual test efforts by tenfold. The automation will consider any number or types of scenarios and can run the same without the intervention of human. This must be done to check whether the developed application is up to our customers needs. This is very important as we must stick to the SRS while developing applications. By integrating proper tools, we can achieve higher efficiency in our testing process. In this study we can see how to integrate Google Pub/Sub with Katalon Studio and how to run Katalon Studio test cases in Jenkins and generate reports for the same.

Keywords: Google Pub/Sub, Jenkins, Katalon Studio, Katalon Run-time Engine.

1. INTRODUCTION

The major issue in today's world is how to handle data effectively. With this increasingly large amount of data, it is important that we come up with a solution to handle it. Powerful analytics is required to handle this data. Similarly, there is large amount of data generated in clinical trials. This data is stored in data lakes, where large number of different types of data can be stored. From where the data can be used as per analyzer's needs. One of the tools to get user data non-synchronously is Pub/Sub. Manually handling Pub/Sub channel is more time taking process, that's where automation comes into picture. By automating the process, it will reduce the testing time by ten folds.

Pub/Sub systems allow disseminating information in distributed systems from several sources called the publishers to different subsets of interested users called the subscribers. Data is produced in the form of publications by publishers. Subscribers express their interests for receiving a subset of publications by issuing subscriptions composed of limitations.[1]

Integrating Pub/Sub channel with the testing framework here we are using Katalon Studio and running the katalon test cases from Jenkins's speedup the process as well as user does not have to visit Pub/Sub frequently this helps in securing the rest of the data from the customers and customers can only see authorized data.

2. Ease of Use

Removing Manual Efforts involved in validating Pub/Sub

The manual intervention involved in comparing the pulled message with the pushed message can be completely removed. It makes the process more efficient and reliable.

Multiple Scenarios can be tested easily by using this approach

Numerous test cases can be specified in an external file which the automation framework must take and provide output in the form of reports.

Multiple Reports can be generated at different places by building the test case once.

By building the test case once in Jenkins, it will generate required formats of reports at multiple places like Katalon Studio, Katalon TestOps, as well as Jenkins.

3. LITERATURE REVIEW

In order to achieve this automation, a framework is designed by integrating Katalon Studio with Katalon TestOps, Pub/Sub, Jenkins, TestNG. The language used for this validation is Java. After proper research on all these tools we can get an idea on how to integrate it into one framework.

Katalon Studio: It is an automation testing solution. This software is built on top of the open-source automation frameworks Selenium, Appium with a specialized web interface for Web, API, Mobile and Desktop application testing.

Pub/Sub: It is flexible, reliable, real-time messaging service for independent applications to publish and subscribe to asynchronous events. As part of the Google Cloud Platform (GCP), google provides Pub/Sub as its queuing mechanism. Once topic is created you can create publishers to publish messages to topic and subscribers to receive messages from topic. For doing all the above you have to create user credentials. Only people with credentials you created can see your published messages.

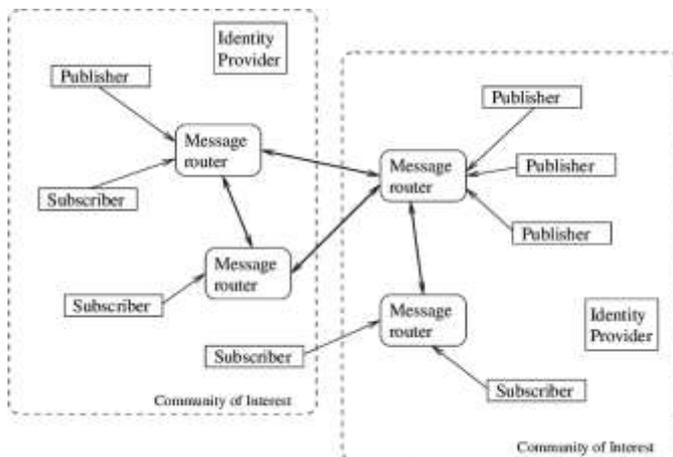


Fig-1: Pub/Sub System Design Outline

Jenkins: It is free and open-source automation server. It aids continuous integration and delivery by automating the elements of software development related to building, testing, and deploying. It is server-based system that run-in servlet containers such as Apache Tomcat.

Katalon TestOps: It is set of cloud-based services to help streamline software quality by continuous test execution and intelligent analytics. Test Planning, Test Management, Reports and Analytics, and Configuration are key modules of Katalon TestOps.

Katalon Runtime Engine: It is test execution add-on of katalon studio. It allows automation tests execution in CLI (Command-line Interface) mode. It can be used in Variety of scenarios like scheduling your tests, integrating with CI/CD system, or bundling your tests to execute in virtual containers like Docker.

4. METHODOLOGY

For Pub/Sub validation we need above tools as specified. The software used is Katalon Studio, which supports Groovy and Java programming languages. Here we are using Java programming language for creating our custom keywords, i.e., for pushing and pulling messages to/from Pub/Sub. To validate Pub/Sub first we have to integrate Pub/Sub to Katalon Studio by adding the required or proper Pub/Sub dependencies to the build.gradle file present in your project file and building it using the command:

Gradle katlonCopyDependencies

The required dependencies are as listed below:

```
compile 'com.google.cloud:google-cloud-pubsub:1.112.3'
compile 'com.google.api-client:google-api-client:1.23.0'
compile 'io.rest-assured:rest-assured:3.2.0'
compile 'io.rest-assured:json-path:3.2.0'
compile 'io.rest-assured:json-path:3.2.0'
```

Once building is done then you have to provide service account credentials to the project that you can do by adding credentials to environmental variables in your local machine.

Now you have to create custom keywords for publishing messages to the Pub/Sub as well as for pulling the messages from Pub/Sub, you can pass the Project ID and topic ID to the custom Publishing keyword to publish the message in specified project and specified topic, and for

pulling messages from Pub/Sub you have to pass project ID and Subscription ID to custom Pulling keyword.

Katalon supports Groovy scripts for creating your own custom keywords, as groovy supports java, you can use java for creating your custom keywords.

For running Katalon Studio test cases from Jenkins first you have to install Katalon Runtime Engine (KRE) and Jenkins for running the test cases in CI/CD mode. Once you install KRE and Jenkins, next you have to install Katalon TestOps plugin in Jenkins to integrate Jenkins with Katalon studio.

Now in Jenkins you can build your test cases by passing the command generated for CI in Katalon Studio along with Katalon Studio path in command window in build environment, but the report will be generated only in Katalon Studio's project folder. To generate reports in Jenkins you have to TestNG plugin and for viewing the reports in graphical manner you have install Test Result Analyzer plugin.

In Jenkins create a new style project with suitable name and in build section select Execute windows batch command option and paste the command generated for console of test suit/test suit collection in the command window with path address of your Katalon Runtime Engine, in post-build Actions select Publish JUnit test Result report option and paste the path address of your XML report file in Test report XMLs window, and click build and save. Now build your Test case.

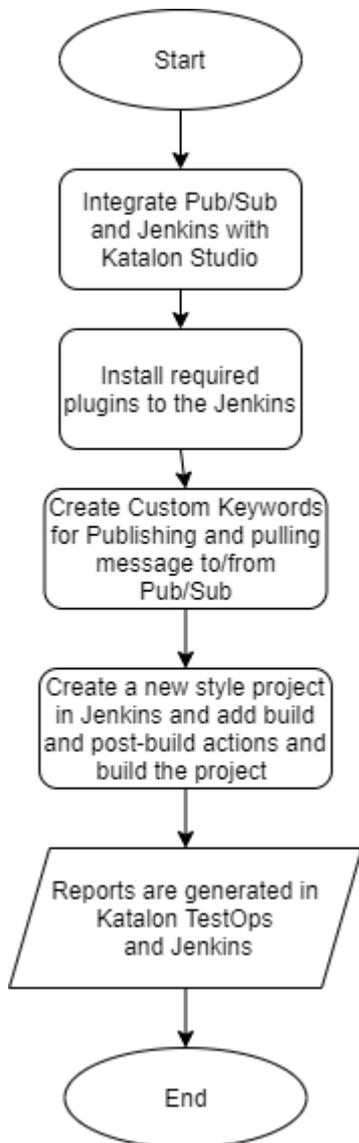


Fig -2: Flow Chart of the System

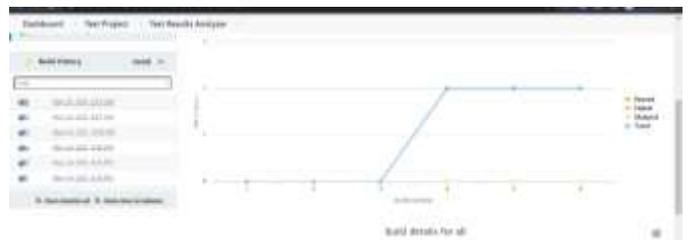


Fig-3: Graphical Report Generated 1

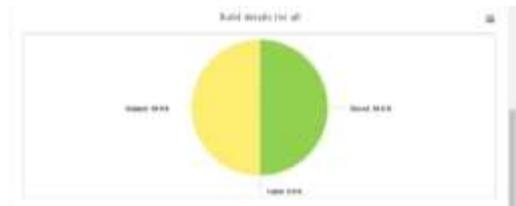


Fig-4: Graphical Report Generated 2



Fig-5: Graphical Report Generated 3

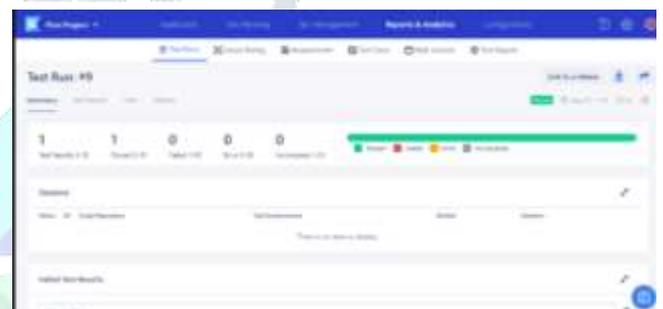


Fig-6: Test Result Generated for Build 9 in Katalon TestOps

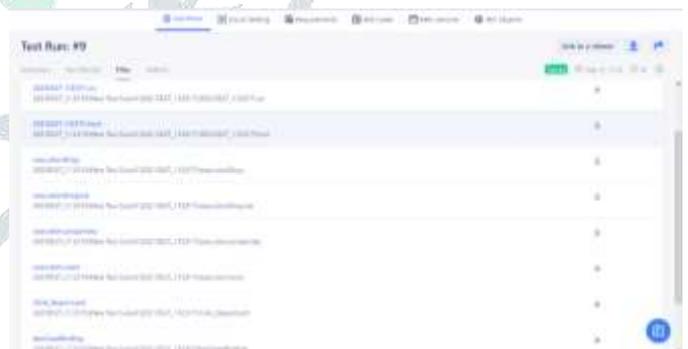


Fig-7: Different Formats of Reports Generated for Test Run 9

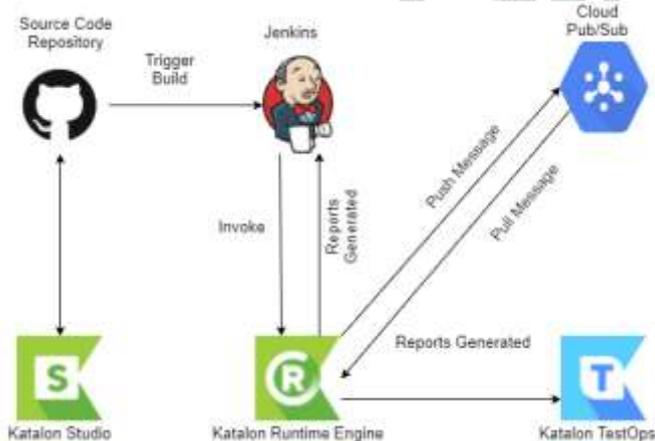


Fig-3: Platform Technology Stack

4. RESULT AND ANALYSIS

The reports are generated at multiple locations with different formats like HTML, XML saying whether the test case is passed or not. If test fails then the reason is displayed to user. Graphical results are also generated in Jenkins telling the number of test cases passed, failed and skipped, which makes users to see the results in a statistical way.

5. CONCLUSIONS

This system successfully works for publishing messages to Pub/Sub and pulling messages from Pub/Sub asynchronously. This can be used for validation of Pub/Sub in testing framework. In this project we successfully generated the reports in required format at multiple locations, which also generates the reports in graphical manner which gives statistical view of the reports.

6. FUTURE SCOPE

This system can be extended for validating the other queuing mechanisms like Kafka by adding the required dependencies and we can also integrate as more frameworks so that users can access all the required data from single place.

ACKNOWLEDGEMENT

We would like to thank our mentor of this project Shankar K, Vinod- Software Engineer II and project manager Arun Kumar-Senior Software Manager at Zebra Technologies and Dr. Mamatha T- Assistant professor at RVCE for helping in completion of the project successfully. They have shared necessary resources for different techniques and methods for this model. We also like to thank to our college committee members and the Head of Department Dr. Ramakanth Kumar P for permitting us to select this project.

REFERENCES

1. <https://arxiv.org/pdf/1705.09404.pdf>
2. <https://docs.katalon.com/katalon-studio/docs/index.html>
3. <https://cloud.google.com/pubsub/docs>
4. <https://github.com/katalon-studio/katalon-gradle-plugin>

