Study on GRAPHQL and Automation Testing

Deviselvam[1], S.Gayathri[2], M.Monisha[3]
[1] *Assistant professor (SG), Computer science and engineering, Sri shakthi institute of engineering and technology*
[2,3] *UG Scholar, Computer science and engineering, Sri shakthi institute of engineering and technology*

***Abstract*- GraphQL is a query language which is mainly used for API. It has been released only as a specification. There are 3 different kinds of architectures which include a GraphQL server, GraphQL server with a connected database and GraphQL server which is a thin layer in front of a number of third party or legacy systems. It integrates them through a single GraphQL API .It is a hybrid approach of a connected database and third party or legacy systems which can be accessed through the same GraphQL API. This gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time which in turn enables powerful developer tools. API testing can be done using automation tools. API testing is a type of software testing that involves testing application programming interfaces (APIs) directly and it is a part of integration testing which is used to determine if they meet expectations for functionality, reliability, performance, and security. As APIs lack a GUI, API testing is performed at the message layer. When we are testing an API, we deal with the stuff under the covers which include JSON, XML, REST, and web services .In order to provide effective and efficient software testing services, IT globally implements various automation solutions. Test automation utilizing Selenium framework helps enterprises to improve their service quality and reduce costs. The Selenium driven framework automates submission of requests and validates the response using a web browser. This reduces significant manual effort and time.**

***Index Terms*- GraphQL, Query, Mutation, API, Testing Tools**

INTRODUCTION

GraphQL is a language that is used for querying the API. It is used for executing queries using the data types that we define for our data. GraphQL can be used to query data from any database that it is currently related to. A GraphQL structure works by providing the required fields and types and the functions that each fields can work on. A GraphQL service is mainly used to provide what the user exactly wants. Earlier RESTful architecture APIs were used that provide the information as a whole. But GraphQL is unique in its own way that it provides user specified information.

Fig.1 GraphQL Logo

PROPERTIES OF GRAPHQL

A GraphQL has two main properties i.e., queries and mutations. A mutation is used to provide inputs. Here the input type can be specified for the inputs. On the other hand, a query is used to retrieve specific data from the database.

For example, take a look at the following GraphQL structure:

```
type sample {
  name:String
}
input sample {
  name: String
}
```

In the above example, type is used to specify the type of data and the input is used the provide input of the given type. Along with that, functions can also be specified for each field on each type.

Once a GraphQL service is running on a webserver, GraphQL queries and mutations can be sent to validate the data that is given as input. The query is

first checked to identify if it is of the required type. If it is not of the given type, it immediately throws respective error messages, else it processes the query to provide the output. For example the query:

```
{
 example {
  name
 }
}
```

Could produce the JSON result:

```
{
 "example": {
  "name": "John Stephen"
 }
}
```

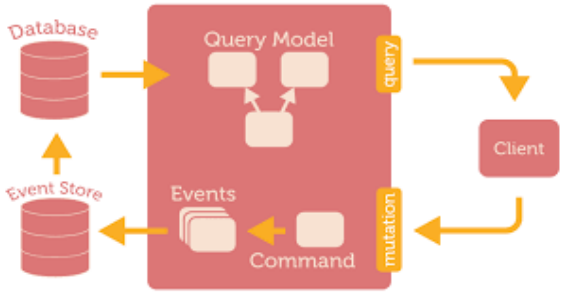

Fig.2 GraphQL process

PROS AND CONS OF GRAPHQL

Representational State Transfer architectures were used earlier for the APIs. This architecture was supported by companies like Twitter and Google. However due to some drawbacks faced in the REST architecture, developers moved on to GraphQL architecture. The central idea behind REST API design is the resource. A typical REST API will comprise a distinct endpoint for each resource, with different HTTP request methods defining how that resource will be interacted with. While this means that REST APIs can become very verbose, standard create, read, update and delete (CRUD) actions can be intuitively composed for each resource. In typical REST parlance, to update an individual resource, you would use the PATCH request method instead of GET. Creating a resource uses POST, and DELETE removes a resource. However not every application can guarantee the use of a CRUD structure, and operations like searching, dealing with large data sets can break organized presentation of any REST API.

Because of the above drawbacks in REST architecture, GraphQL is becoming one of the mostly used architectural style in recent days. In contrast to REST's highly structured, resource-driven design, GraphQL acts as a more flexible query language on top of an API. This allows normal users to interact with an API in a very programmatic way, rather than the more rigid REST API black box.

AUTOMATION TESTING

Automation testing is a technique which is used as an application to implement entire life cycle of the software in less time. It provides efficiency and effectiveness for testing software. Automation testing is an Automatic technique where the tester writes scripts by own and uses suitable software to test the software. It is basically an automation process of a manual process. As regression testing, Automation testing is also used to test the application from load, performance and stress point of view. In other words, Automation testing uses automation tools to write and execute test cases. It involves no manual effort which is required while executing an automated test suite. In usual way testers will write test scripts and test cases using the automation tool and then finally group them together into test suites. The main goal of Automation testing is to increase the test efficiency and to develop software value. Automation tools are used to write and execute test cases which are known as automation testing.No manual intervention is required while executing an automated test suite. This differs from manual testing where a human being will be responsible for single-handedly testing the functionality of the software in the way a user would. Because of automated testing done through an automation tool, only less time is needed in exploratory tests and more time is needed in maintaining test scripts which in turn increases overall test coverage. The main benefit of manual testing is that it allows a human mind to draw insights from a test that might be missed by an automated testing program. Automated testing is well-suited for large projects. It is mainly used for projects which require testing the same areas over and over. This is also used for projects which have already been through an initial manual testing process. Automation Scripts are executed during this phase. The scripts need input test data before there

are set to run. Once the test cases are executed, they provide detailed reports which can be used as test report.

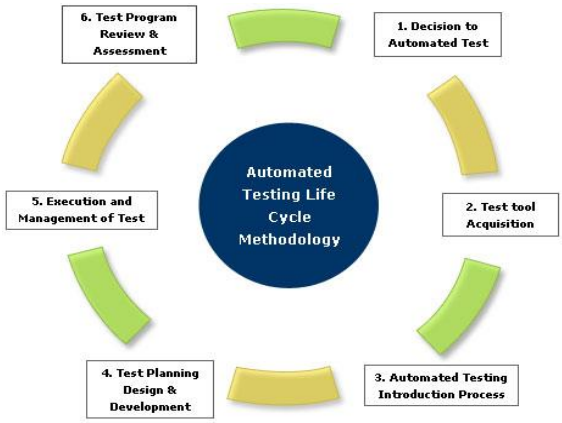

Fig.3 Automated Testing Life Cycle Methodology

Execution could be performed using the automation tool directly or through the Test Management tool which invokes the automation tool. Scripts which can be executed in a single machine or a group of machines can be done during night in order to save time. Test automation is a process to check the software application after development and in order to get a new build or release. The investment for test automation is time, money and resources. The type of testing (manual or automated) depends on various factors, including project requirements, budget, timeline, expertise, and suitability. There are three vital factors of any project which includes course time, cost, and quality. The main goal of any successful project is to reduce the cost and time required in order to complete it successfully while maintaining quality output. When it comes to testing, one type might be accomplished with this goal better than the other. While testing your demands that you run the same set of test cases simultaneously on more than one machine, then you need to use automation testing. With manual testing, you cannot type the same test cases to run exactly at the same time on several machines. However, with automation testing, where you can schedule the scripts so that the test cases are executed exactly at the same time on more than one system. Automation testing is the application of tools and technology to testing software with the goal of reducing testing efforts, delivering capability faster and more affordably. It helps in building better quality software with less effort. Many companies are already using automation testing to a certain extent, but still largely depend on manual tests because they don't know how to properly leverage the benefits of automated testing in their development process.

Manual testing is performed by carefully executing predefined test cases, comparing the results to the expected behaviour and recording the results. Manual tests are repeated each time the source code changes and is prone to errors. It is also difficult to execute on multiple platforms. It is necessary to invest significant time and effort when introducing automated tests in an organization. However, there isn't much of a financial commitment, at least not while starting out on a small scale. There are numerous open source test automation tools that could be made use of, especially in the early stages.

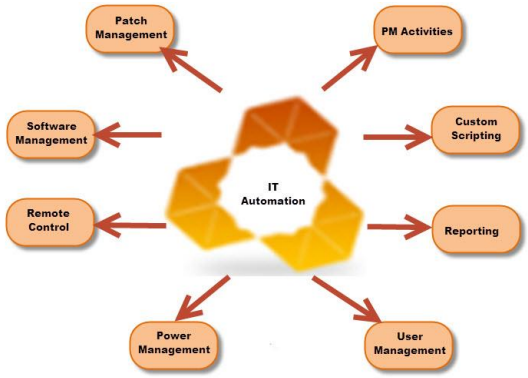

Fig.4 IT Automation

STUDY ON AUTOMATION TESTING TOOLS AVAILABLE

The increased demand for automation is trending in our software testing industry. There are different types of testing such as desktop testing, web testing, browser testing, regression testing, web services and API testing and many more.

Top 5 Testing Automation Tools that are available for Software Testing are

Selenium

Selenium is a testing framework which is mainly used to perform web application testing across various browsers and also platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl. It offers record and playback features which supports to write tests without learning Selenium IDE.

Selenium as a native part of their browser is a famous open source testing tool available. Selenium is

undoubtedly the base for most of the other software testing tools in general.

TestingWhiz

TestingWhiz is a test automation tool with the code-less scripting by Cygnet Infotech, a CMMi Level 3 IT solutions provider. TestingWhiz tool's Enterprise edition offers a complete package of various automated testing solutions like web testing, software testing, database testing, API testing, mobile app testing, regression test suite maintenance, optimization, and automation, and cross-browser testing. TestingWhiz offers various important features which include Keyword-driven, data-driven testing, and distributed testing. It also offers record and playback test automation framework. This includes object Eye Internal Recorder and 290+ inbuilt testing commands in addition to in-built JavaScript. It can be integrated with bug tracking tools like Jira, Mantis, and FogBugz. Then it can be integrated with test management tools like HP Quality Center. Risk-based testing mainly includes continuous Integration and delivery in Agile cycles.

HPE Unified Functional Testing (HP – UFT formerly QTP)

HP QuickTest Professional was renamed as HPE Unified Functional Testing. It offers testing automation for functional and regression testing. It includes Visual Basic Scripting Edition scripting language which is mainly used by this tool to register the test processes and it operates the various objects and controls in testing the applications. QTP offers various features like integration with mercury business process testing and mercury Quality Center. It consists of unique smart object recognition. It includes error handling mechanism such as creation of parameters for objects, checkpoints, and data-driven tables. It provides automated documentation.

TestComplete

TestComplete is a functional testing platform which offers various solutions to automate testing for desktop, web, and mobile applications by SmartBear Software.

TestComplete offers the features such as GUI testing. It supports scripting languages such as JavaScript, Python, VBScript, JScript, DelphiScript, C++Script and C#Script. It also supports test visualizer, Scripted testing and supportstTest recording and playback

Ranorex

Ranorex Studio offers various testing automation tools that cover testing all desktop, web, and mobile applications.

Ranorex offers features such as GUI recognition, reusable test codes, bug detection, integration with various tools and it supports record and playback

AUTOMATING GRAPHQL API TESTING

Automated testing is an extremely useful as a bug-killing tool for the modern developer. You can use a collection of tests or a test suite to solve or avoid a number of problems.

Testing a GraphQL application is a complex task, because a GraphQL application is made of several layers of logic – schema definition, schema validation, permissions and field resolution. With Graphene test-execution framework and assorted utilities, you can simulate GraphQL requests, execute mutations, inspect your application's output and generally verify your code.

GraphiQL

Many GraphQL APIs use this open source console as an interactive API playground. GraphiQL is the popular Integrated Development Environment (IDE) for interacting with GraphQL API calls, enabling developers to query data and perform mutations. This IDE is relatively easy to implement for Node.js servers, express-GraphQL can automatically generate GraphiQL. Since it's built on React, GraphiQL can also be injected with unique CSS for custom branding.

Fig.5 GraphiQL

GraphQL Voyager

GraphQL Voyager can make for a cool experiment. Voyager takes a GraphQL API and turns it into a visual graph.Then after setting a root schema, you

can visually view how fields are connected to types. Voyager is interactive, selecting a type highlights the fields it is comprised of and links to relevant data within the graph.

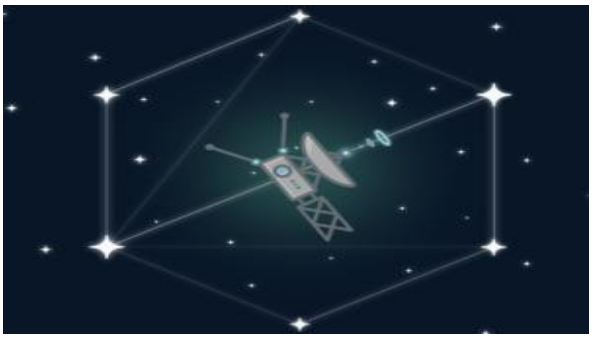

Fig.6 GraphQL Voyager

GraphCMS

GraphCMS is an API-centric Content Management System (CMS) that is intimately tied with GraphQL. It lets you to build a hosted GraphQL backed for web apps, providing tooling to manage content. Users define data structures, validate them in a GraphQL console, and can see the representation in the user interface within the same platform. While GraphCMS is not a good fit for an existing API platform, it would take over well to a blog, web app, or other content structures that require the ability to programmatically share data. A GraphQL-based CMS would be an interesting alternative to traditional CMSs like WordPress or Drupal, and would enable a more futuristic content management framework that comes API-equipped and it is thus a more flexible management layer for end user interfaces.

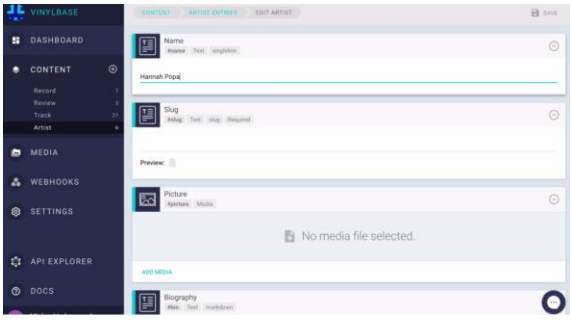

Fig.7 GraphCMS

Graphcool

With flexible, scalable, serverless architecture the rage, having a pre-formed backend is an interesting prospect, especially when it is GraphQL compatible out of the box. Graphcool is a platform to aid GraphQL schema design and app backend development which is coming with a visual console to design and edit your data schema. With the ability to create advanced GraphQL data models, we can custom fields and have relations between data. Integrations with many popular tech like AWS Lambda, Algolia, and Auth0, Graphcool looks to be a powerful tool for modern database management.

Optics by Apollo

To round off our list of GraphQL tooling, Optics is a product for monitoring GraphQL APIs. Optics is an analytics solution for GraphQL APIs that traces how queries is running and it helps you see what types of queries are being performed and their frequencies. It consists of API metrics which are crucial and any web API can benefit from adding an analytics solution to their platform. Seeing request volumes displayed visually, as well as having a better grasp on bottlenecks like latency issues is necessary for optimizing performance and improving page load times.

PROS AND CONS OF AUTOMATION TESTING

Pros of automation testing

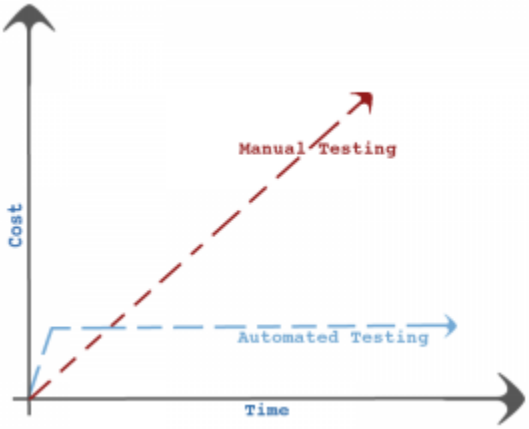

Fig.8 Automation Testing Trend

A. Fewer human resources

This requires only fewer human power to test the solution. A single automation engineer can take hold of writing the scripts and automating the tests.

B. Reusability

The scripts are reusable:

New scripts need not be written each and every time even if the version of OS on the device is changed.

The test can be redone without any changes. Automation is very helpful in finding bugs in early stages of software development so that these bugs can be fixed in early stage itself. This also results in reduced expenses and reduced working hours to fix these problems.

C. Reliability

Automation testing is more reliable and helps save time when repetitive standard tests are to be carried out which cannot be skipped. It also helps identify bugs which cannot be identified when manually tested.

D. Simultaneity

You can test more devices simultaneously resulting in comparative detailed reports generated in less time with the exact the same parameters, because the exact same scripts were run.

E. Continuity

Automation Engineers can have a clear picture of what exactly the other engineers have done, what scripts they have already written and what tests have already been performed and what bugs were already found and fixed, through clear reports. To achieve the best results a combination of both types of testing is required: automated testing for repetitive, simple use-cases; and manual testing for reproducing specific bugs, complicated use cases and to ensure the best user experience possible. Through the solutions that are being offered, one can guarantee that the testing needs can be satisfied.

Cons of automation testing

Though the automation testing has many advantages, it has its own disadvantages. Some of the disadvantages are:

A. Automation script usages

Writing automation test script is an art. It should be written in such a way that it provides appropriate test results for any kind of inputs. Debugging the test script is major issue. If any kind of error is encountered in the test script, it may lead to unexpected consequences

Maintenance of test data files is difficult, if the test script tests more screens. Automation Testing can be expensive for smaller companies with smaller projects. Those companies that cannot afford for automation testing rely on Independent testing companies like test bytes. Manual Testing cannot be replaced: Though Automated tools are faster and precise, they have some limitations compared to Manual Testing. For example, they just do what they are programmed to do. Hence they cannot replace manual testing.Automated testing fail in compatibility since all tools do not support all UI objects and testing. Some tools are specialized for UI Testing, some for DB Testing, some for web testing, and performance/load testing.

TRENDING TECHNOLOGIES IN AUTOMATION TESTING

The last decade has seen an overwhelming growth of the software testing industry. This rapid scale of development is providing an enormous energy not just for the developers, but also for the testers making them to continuously strive to enhance their skills. Even businesses today need to have a look or be even more aware of what is best in terms of performance and security. This situation has been caused by the new technologies, and it is appears to be a challenge for testers to overcome the new issues posed by these upcoming technologies. Support communities for the open source tools can only become more and more involved and active. Quality assured with High speed is the new requirement of almost all business: Everyone wants the best products in the fastest possible time. This urges organizations to focus on providing the best user experience along with the fastest time to market. Software Development Engineers in Test (SDETs) have been existing among us since almost a decade, but their role was very different from traditional testing roles. That in turn specifies, by early 2020, almost all testers will need to wear an SDET hat to be successful in the field of Test Automation, that is going to play a major role in the world of testing. Agile and DevOps will rule the roost – TCoE is dead: According to Forrester, organizations are not looking at having centralized Test Centres of Excellence anymore. The erstwhile testing arena is making a shift towards quality engineering, and testing is intended to become more iterative, progressive, and seamlessly integrated with development. Digital Transformation is here to play a

major role: With a majority of organizations making a entry in the digital world, the need for digital transformation will be redirected to a huge shift of focus towards digital testing. .BigData Testing will become massively BIG: We are sitting on top of an explosive amount of BigData today and need to have a very strong strategy around BigData Testing. Testing datasets requires highly analytical tools, techniques, and frameworks, and is an area that is set to grow big.

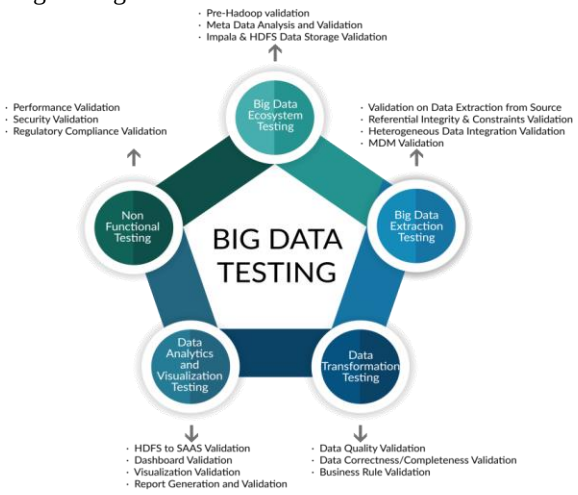

Fig.9 Big Data Testing

Internet of Things Testing

The IT industry is witnessing a tremendous change where we see devices that are connecting everything. Yes, it is the Internet of Things that is going to rule the world and it is going to be among the biggest industries by the 2020. So here comes the IT testing into picture. As there are more IT devices in the market now that are being used by masses, there is a need to have better IoT Testing to offer better software product quality. Software development companies across the globe are creating software product strategy that primarily focuses on IoT development and its testing.

Performance Testing is the New Thing

Alongside performance engineering, there are speculations that it will soon be replaced with Performance testing. Here, when we say performance testing, we mean that testing that ensures user experience which is the most important factor in software product. It is really important to have consistent performance across platforms, operating systems, to know the efficiency of the product. This derives the success of the software product.

Software Testing is now ever growing

The IT industry has surpassed the era of manual software testing and slowly the focus is shifting to Automation Testing. And so it is necessary for software testers to upgrade their skills and also consider testing for various new innovations like IoT and BigData and even more. Yes, software testers and team of SDETs at SwissHelios are already learning tools and techniques to improve software testing and take software solutions to next level.

CONCLUSION

GraphQL is a syntax that describes how to ask for data and it is generally used to load data from a server to a client. It has three main characteristics as it lets the client specify exactly what data it needs and makes it easier to aggregate data from multiple sources as it uses a type system to describe data. Automated software testing will increase the depth and scope of tests which helps to improve software quality. Usually lengthy tests which are often avoided during manual testing can be run unattended. They can even be run on multiple computers with different configurations. In order to achieve the best results we should use combination of both types: automated testing for repetitive, simple use-cases and manual testing for reproducing specific bugs, complicated use cases and to ensure the best user experience possible.

REFERENCES

[1] http://graphql.org/
[2] http://graphql.org/learn/queries/
[3] http://searchmicroservices.techtarget.com/tip/GraphQL-vs-REST-Nailing-down-the-pros-and-cons
[4] https://dzone.com/articles/top-10-automated-software-testing-tools
[5] https://nordicapis.com/10-awesome-tools-and-extensions-for-graphql-apis/
[6] www.cigniti.com/.../10-emerging-trends-in-software-testing-predictions
[7] https://www.infoworld.com/article/.../better-software-testing-through-automation