

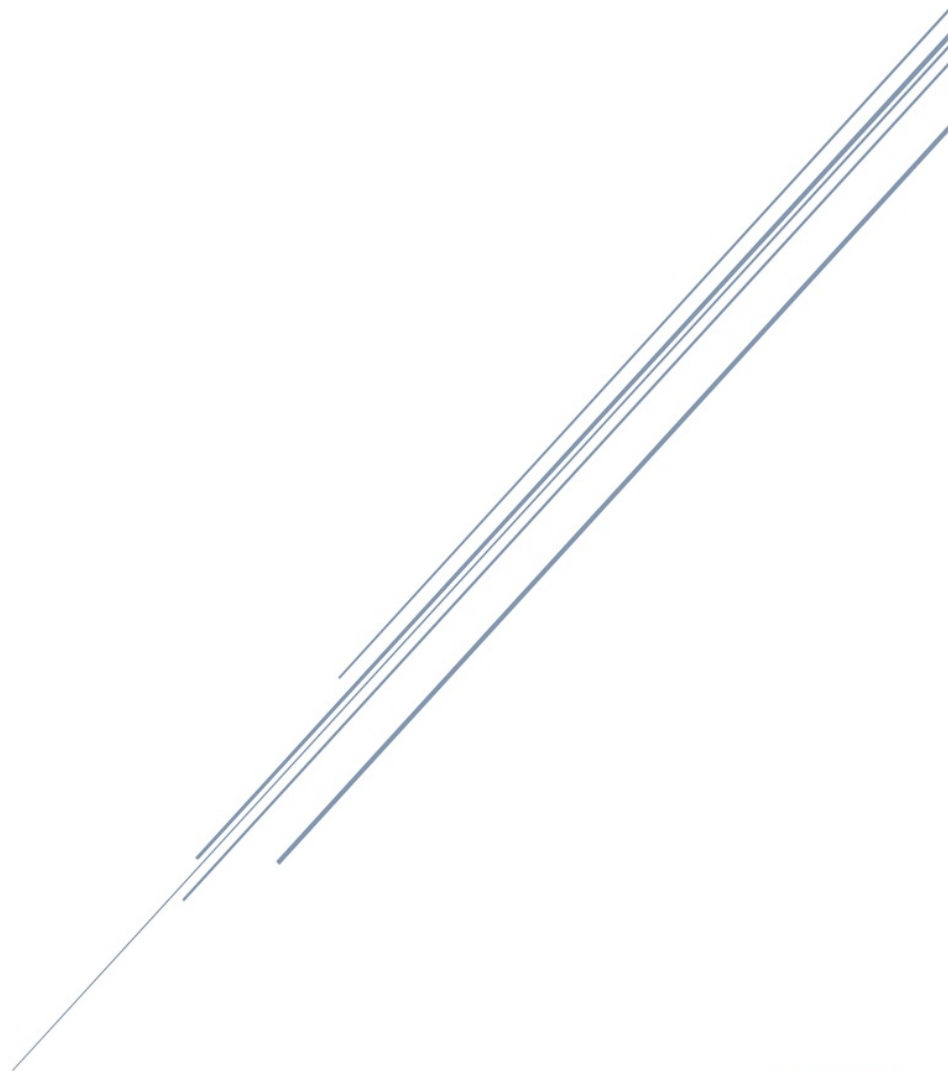
exploringEmergingWebTech

by Thomas Evans

FILE	2770_THOMAS_EVANS_EXPLORINGEMERGINGWEBTECH_178152_135890639.DOCX (336.13K)		
TIME SUBMITTED	04-APR-2019 03:50PM (UTC+0100)	WORD COUNT	1863
SUBMISSION ID	104084464	CHARACTER COUNT	9469

INVESTIGATING EMERGING WEB TECHNOLOGIES

Mike Dacey



UWTSD
Web Development

Introduction

This report will be split into 2 sections. The first section will explore an exciting upcoming web technology that could potentially have a large impact on the future of web development. Later in this section there will be an evaluation of a NoSQL database. Once the evaluation has been complete a description of the technologies possible use will be completed as well as a comparison of SQL and NoSQL with the benefits and drawbacks of using NoSQL as oppose to the traditional SQL database.

In this second half of the report, the report will cover the testing of an application to accompany this report. The application must utilise typescript and AJAX to access a rest web service and can also incorporate bootstrap. This section will cover the testing of the applications features. The application should also contain demonstration code for the web technology which should be integrated into the application.

Knowledge and understanding Skills

GraphQL

This section will cover an emerging web technology that the author of this report believes will have a large impact on the future of Web Development. For this section of the report the technology being covered is GraphQL. GraphQL is an open-source query language for communicating with API's. The technology was developed by Facebook in 2012 and was released formally in 2015. GraphQL was moved from Facebook to a newly established GraphQL foundation in 2018. The foundation is hosted by the Linux foundation. GraphQL is both a query language and a specification for an API Service.

This section of the report is now tasked with evaluating the technology and discussing why the author of this report feels that it will create a large impact on the future Web Development.

The existing technology that GraphQL wishes to expand on is the REST API technology. REST (Representational State Transfer) API implantation allows clients to create, change or gain access to data from a web service via a set of defined end points for the transaction of data [1]. Although this technology is extremely useful when using or creating web services it does present an element of inflexibility. In order to complete a data-request the service must have an available end point for the data required. This can make it necessary for REST API interfaces to be rewritten in order for the API to meet the client's needs. Even if an API evolves to meet the needs of all its component's then some of them might only require a portion of the data fields or a combination of fields from different objects. This may result in none of a REST API's endpoints being able to satisfy their needs and may lead to a user having to make several requests to different end points in order to gather all the data they need.

GraphQL was created in order to solve the issues that REST API's have. GraphQL wishes to improve the flexibility and maintainability issues that are apparent when using REST API's. "The goal of GraphQL is to unify data request of an (eco)system into one endpoint, reducing the amount of requests needed to retrieve data." [1] This is why the features of GraphQL makes the language suitable for use in data-driven technologies as it can provide flexibility, interoperability and help improve a systems efficiency and performance by reducing the number of requests necessary for retrieving the desired data for an application's requirements.

In their paper [2] wished to study GraphQL as an alternative to REST API. Their goal was originally to gain a fundamental understanding of the language's properties. In their conclusion they state that the language is becoming increasingly popular and an alternative to the REST-based API's. They also state that the language used a schema to describe the data's organisation and a declarative query language in order to allow clients to access data. They state that they studied the complexity of the language and in turn studied the problems the language had with evaluation, enumeration and size-computation. They stated that through their study that these issues could be solved efficiently.

In their paper [3] describe the use of GraphQL in their system. The paper describes the transition for the EHRI to make their data accessible using the GraphQL. The service is offered as a database an extension and their paper covers the implementation of the extension onto their service. They did cover some of the limitations of the language by stating that it prevented them from integrating a full-text search engine into the API. They do however go on to state that although this is a drawback of using it. With this preventing them from expanding it into a single endpoint. That the benefits of using the language outweighed the negatives of using it in combination with their EHRI portal.

As stated EHRI's use of the GraphQL language is an example of where the technology has been successful. The website for the API states "The GraphQL implements Facebook's data query language and provides the means to make highly specific queries of the portal's data with a wider scope than that provided by the search API" [4].

No SQL

The following section will cover NoSQL databases. This section will cover and evaluation of key value NoSQL databases and will discuss an example of its use. Later in this section will be an analysis of NoSQL and will cover the advantages of using NoSQL as oppose to traditional SQL databases. Traditionally there are 4 types of NoSQL databases key-value store, Document-based store, column based store and Graph based.

A key-value database is the most flexible variant of NoSQL databases. Key-Value databases have emerged as a viable alternative to the traditional relational database. Key-Value databases store data using a key value. A key value being a unique string value that belongs to each object inside of the array of the database. Unlike traditional relational databases key-value NoSQL does not restrict the content of the database by type. This form of No-SQL database does allow users to access data for reading and storing inside memory to enable easy access when needed.

The key field in this kind of NoSQL database can be either auto-generated or synthetic and can consist of string, JSON or BLOB.

A few examples of key value store being used could be a phone directory where each object would contain a unique name "Bob" and would then contain their phone number "07394552532". Although this example does fall apart if the user has more than one contact with the same name as the key fields must be unique. Another example could be a database containing IP address. In this scenario the IP address could be used as the key field and the MAC address could be us stored as the value. The database could also be applied to an ecommerce application where each item has a stock number (Key) and a description (Value).

A case study developed a NoSQL database to replace their original relational database due to their growth of data. The case study in this scenario is health care institute that covers data for nine million patients over 100 facilities [5]. They stated they their data growth was around a terabyte a month. And that all data they had must be retained for a minimum of 99 years before it can be deleted.

When using NoSQL databases there are some advantages and disadvantages to using them over your traditional relational databases. Firstly, NoSQL databases are far more suited for the use of Big Data applications. Although relational databases can hold large volumes of data it is realistically easier to use NoSQL databases as they can easily be installed in a cheap commodity for larger volume databases. This also shows another advantage in the economy of using NoSQL over relational databases.  1

On the other hand, there are some disadvantages to using NoSQL databases as oppose to its counterpart. For example, as NoSQL are only recently growing in popularity there is less support for the technology. With most NoSQL databases being developed by smaller start-ups which will lack resources in comparisons to the established vendors of relational databases like IBM or Microsoft. By extension of this there is a lot less advanced expertise when it comes to using the NoSQL databases due to the lack of time they have been popular. Although these are major disadvantages now it is likely that with the growing popularity of the technology that they will not be a problem after the technology has grown after few years.

Practical Skills

This section of the report will discuss the application created to accompany this report. The application created utilises the open weather API in order to request information about the weather in a specific location. For this purposes of this report the destination chosen will be Swansea. Below is testing table that will cover a features expected outcome and evidence of testing to see whether it works as expected.

Feature	Expectation	Evidence	Actual
Website should receive data from the API once called with the specific location.	The console tab on develop tools window should display an array of data that from the API		The console window contains the information from the api.
The application should display an icon for the weather.	The API should grant the application access to the image for the corresponding weather type.		The weather icon is featured on the web page
The the description for the weather and the temperature should displayed on the website.	The website should request the weather description and the temperature from the API and display it on the web page.		The web page displays both sets of data on the web page. The API originally provided the temperature in Fahrenheit so a conversion needed to be made and rounded up .before displaying and the symbol for degrees Celsius added.
The website should allow the user to interact with API.	The webpage should allow the user to select , add , edit and delete content		The application does not allow the user to interact with the API as the API chosen does not allow users to edit the data from the API.

GraphQL

The brief for this report demanded that the new technology explained in the first part of the report be implemented in the application created in some form. Sadly, due to the technology being new the author of this report was unable to find a GraphQL that would have made sense to include in the web page. Therefore, it is impossible to include any features of the technology in the webpage.

References

- [1] Andrea Vázquez-Ingelmo, Juan Cruz-Benito, Francisco J. García-Peñalvo, (2017), Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL, October 2017
- [2] Olaf Hartig, Jorge Pérez (2018), Semantics and Complexity of GraphQL, April 2018. International World Wide Web Conferences Steering Committee
- [3] Mike Bryant (2017), GraphQL for Achival Metadata: an Overview of the EHRI GraphQL API, 2017. 2017 IEEE International Conference on Big Data (Big Data)
- [4] Portal.ehri-project.eu. (2019). *EHRI - EHRI Data APIs*. [online] Available at: <https://portal.ehri-project.eu/api> [Accessed 3 Apr. 2019].
- [5] John Klein, Ian Gorton, Neil Ernst, Patrick Donohoe, Kim Pham, Chrisjan Matser,(2015), Performance Evaluation of NoSQL Databases: A Case Study. February 2015.

FINAL GRADE

50/100

GENERAL COMMENTS

Instructor

Eval emergent tech (20) 18

Eval of NoSQL (30) 22

Typescript application (20) 6

REST web service (20) 4

Tech demo (10) 0

The emergent technology that you chose was GraphQL. An interesting approach to a problem I was unaware existed, and you have provided a good explanation.

You didn't choose a specific NoSQL database, instead you provided a general discussion. As a result, your evaluation was a bit vague, but you did provide some examples where the NoSQL approach could be adopted.

Code has been provided but there is just a small amount of JS code for fetching data from an existing service. Obviously, there is no Typescript or REST service, which means the code is very limited. I will give you some marks for the code provided.

Some evidence of testing.

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

NoSQL can also be the best approach if handling unstructured data.