

Assignment 1

by Michael Rhys Jones

Submission date: 10-Dec-2021 03:29PM (UTC+0000)

Submission ID: 165789951

File name: 31100_Michael_Rhys_Jones_Assignment_1_382340_430262341.docx (86.73K)

Word count: 1761

Character count: 9262



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Cloud-Based Web Services

Assignment 1

Michael Rhys Jones

Michael Dacey and Tim Bashford

Table of Contents

Introduction	3
Comparing and Contrasting Web Servers	4
SOAP.....	4
REST.....	4
GraphQL	5
Debate.....	5
Advantages and Disadvantages of SOAP	5
Advantages and Disadvantages of REST	6
Advantages and Disadvantages of GraphQL.....	6
Video Presentation Code	6
References	7

Introduction

The main part of my assignment I to create a dynamic webpage. The webpage will need to use AJAX and REST API in order to exchange information with many other webservices. To allow me to get the marks I need, I will be loading data in a JSON format to allow me to display data while the user is browsing through the web page.

Throughout the next part of the assignment, I will be writing a report. Throughout this report, I will be making thorough research in different messaging protocol languages. I will be studying SOAP, REST, and GraphQL web services to allow me to complete this assignment. To kick start the report, I will explain what each web services are, and explain what they do. I will be comparing and contrast each of the services given to me, and discuss the benefit and drawbacks are for each of them.

Finally, I will provide my work with a 10–15-minute video of me going through my webpage. The video will include a preview and the content of the webpage, and a description of how each section of the web page to show how it works by using code.

Comparing and Contrasting Web Servers

SOAP

SOAP stands for 'Simple Object Access Protocol'. SOAP was a common messaging protocol during the early noughties. It uses XML as a format with a well-defined schematic such as HTTP or TCP. Despite it being used today, people have started using different protocols such as RPC (Remote Procedure Call). SOAP is a messaging XML-based protocol that would be used to examine many network services using HTTP. SOAP has a variety of abilities that can be used through many web services.

SOAP was developed to allow the XML to progress XML with HTTP. This would allow them to be compatible with many web services, and allow these services to talk to different services. XML is known as the 'underlying medium for data exchange'. Without SOAP, XML would not be able to create a messaging body to send off to different services. [1]

The development of Soap was originally designed to allow users to make a foundation for their applications, and websites, and to start building on top of it to allow them to make progress of their prototype design. This was some sort of 'life hack' they would use to help them make their build. This would allow them to put in less effort into the work, making them save time as they wouldn't have to worry about making mistakes at the start of their project. [2]

While SOAP would be standardized protocol, REST would be the architectural side of web services. Meaning that a web Server would contain SOAP, but can also contain REST, but they would perform different activities. SOAP would contain few coding than REST. This would make it easier for a programmer to structure their code as there is less work involved while coding. Due to this, SOAP would be valuable more to use for asynchronous web pages. One of the differences between SOAP, and REST would be the state services they provide. SOAP would be Stateful because it would keep track on changes within a database, or web services, such as transactions etc. REST would be known as stateless because they will need to depend on client processes to gather properties within the web services. This would be the Application State, or the Resource state. [3]

REST

This is another web service that would be used to make an architectural style for API. This architectural style would access and transfer data by using HTTP requests. An example of using REST would be when the web service would use commands such as;

- GET
- PUT
- DELETE etc.

It would use commands like these for performing procedures to create a database etc. It would read, create, delete and more. [3]

Rest would be ideal for programmers as it is fully flexible to use. Meaning that data is not tied down, and can be transferred easily during the process of exchanging data between 2 services. When a programmer would use Rest, they have many constraints to think about;

- Client Server
- Stateless
- Cache
- Uniform Interface
- Layered System

- Code on Demand.

These constraints would have different uses when making code. However, while SOAP is a constraint for XML, REST isn't. On the Other hand, REST can be useful for different structural data format such as JSON and many other kinds of formats. Just goes down to what the client would ask for. [4]

GraphQL

GraphQL is used to allow clients get an accurate data request they ask for. GraphQL will gather every bit of information in a single request from the client side. After a thorough research, GraphQL is an 'upgraded' version of REST. REST would need many URLs to allow information to be past between web servers. Many popular companies use GraphQL due to its high expectations and compatibility. Originally made for Facebook, but GraphQL was designed for other web services to use to upgrade their performances. [6]

With GraphQL and REST being similar, they do have many differences. These differences would mainly be the speed and performance. GraphQL would be more client-driven than Rest because REST API would focus more on the server side of web servers. One of the biggest improvements would be data fetching. REST would contain more endpoints. Meaning that it would have to travel further than GraphQL to get to the other end of the server. GraphQL would only need one endpoint to deliver its data.

GraphQL has a massive impact on error handling for client servers. REST would only display HTTP status codes that not everyone would know. 404 etc. However, GraphQL would take an extra step ahead and display the error reasoning as well as the code next to it.

Like GraphQL, SOAP only goes to one endpoint to deliver information. However, SOAP would require a larger packaging to deliver data to client servers. This could make a network slow, and increases the possibility to crash. This is why GraphQL would be more suitable to use, as it would use smaller packages to deliver different data types. [7]

Debate

Advantages and Disadvantages of SOAP

There are many reasons why developers would still use SOAP today. I believe SOAP has many advantages for its use. SOAP Would be compatible with many online web service environments. This would mean that SOAP can deliver a large quantity of XML terminology for the client's needs. Every Web Services will need a computer language that would need to make communication methods that would go back and forth between the web services and the client side services. SOAP would be the easiest protocol to use for this. Not only that, but SOAP would work on many different platforms, and languages. In addition to the advantages, SOAP has an increase of security. It would encrypt all transactions making it private from unauthorised users.

When a developer uses SOAP, the protocol has the ability to provide the user with extra information telling them where and what the errors, they have made, are. SOAP has a built-in error handler that can notify clients about any possible errors. [8]

SOAP is known to be more heavy-weight than other web services. When Using SOAP, it is required that the user would be knowledgeable, and experienced because it would require many separate tools making it harder to use when developing. This web service would show less performance, and less flexible when being used.

Advantages and Disadvantages of REST

REST is known to be simpler than SOAP. This would mean more developers would be able to use this kind of web service. REST can be used on a variety of formats, just like SOAP. It can also recycle different HTTP tools, meaning that the web server is cache friendly. Beginners would be able to use this service as it requires less tools.

REST has a lack of security. Unauthorised users would be able to decode encrypted transactions quicker than the ones on SOAP. Another disadvantage of this would be that REST can fail to keep its current state. This would make it more complex for beginners. [9]

Advantages and Disadvantages of GraphQL

GraphQL is a lot more complex but it's one of the best web servers to use. GraphQL works by 'less talk' and 'more work' procedure. Instead of going to multiple end points, it only needs to go to one with all the data it needs. GraphQL is capable of sharing code as fragments. Fragments are able to allow you to get different information while keeping the same fields. For the developers, they would receive the usual error messaging if there is a fault within the service. However, the message would be detailed so much, that it would already tell the user how to resolve it.

The chances of GraphQL's performance to slip is very high. If the user has many fields open, the GraphQL would start to encounter performance issues. Even though GraphQL overpowers the other web services, a file upload feature doesn't exist for it. There is a way to upload files but there isn't a specific button to select to do it. This would mean that further training for this would be required to use before proceeding with it. Understanding this service would be difficult to do in a short space of time. [10]

Video Presentation Code

https://youtu.be/alvU_3sxuQM

References

- [1] Ebase, "XML Concepts," [Online]. Available: <https://windmz.dartford.gov.uk/ufs/doc/xml.htm>.
- [2] "SOAP Web Services Tutorial: What is SOAP Protocol?," Guru99, 8 October 2021. [Online]. Available: <https://www.guru99.com/soap-simple-object-access-protocol.html>.
- [3] "REST vs .SOAP," SearchAppArchitecture, 2021. [Online]. Available: <https://searchapparchitecture.techtarget.com/definition/RESTful-API>.
- [4] A. S. Gillis, "REST API (RESTful API)," SearchAppArchitecture, September 2020. [Online]. Available: <https://searchapparchitecture.techtarget.com/definition/RESTful-API>.
- [5] "What is RESTful API?," MuleSoft, 2021. [Online]. Available: <https://www.mulesoft.com/resources/api/restful-api>.
- [6] "GraphQL: Core Features, Architecture, Pros and Cons," Altexsoft, 23 March 2019. [Online]. Available: <https://www.altexsoft.com/blog/engineering/graphql-core-features-architecture-pros-and-cons/>.
- [7] P. Logan, "REST, SOAP, GraphQL — Gesundheit!," Medium, 22 September 2018. [Online]. Available: <https://medium.com/better-practices/rest-soap-graphql-gesundheit-6544053f65cf>.
- [8] P.A. Castillo, J.L. Bernier, M.G. Arenas, J.J. Merelo, P. Garc'ia-S'anchez, "SOAP vs REST: Comparing a master-slave GA implementation," 2011.
- [9] "Advantages and Disadvantages of REST API," Krify, 28 October 2021. [Online]. Available: <https://krify.co/advantages-and-disadvantages-of-rest-api/>.
- [10] "GraphQL: Core Features, Architecture, Pros and Cons," Altexsoft, 23 March 2019. [Online]. Available: <https://www.altexsoft.com/blog/engineering/graphql-core-features-architecture-pros-and-cons/>.

Assignment 1

GRADEMARK REPORT

FINAL GRADE

67 / 100

GENERAL COMMENTS

Instructor

Ajax & REST (45): 21

Written Research (45): 39

Demo (10): 7

Feedback:
=====

The report is a little confusing, although broadly it discusses the main differences. It could have been slightly clearer when discussing the benefits/drawbacks.

The code contains a number of problems, basically because of fundamental miss understandings.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7