

Assignment 1

by Daniel James Bennett

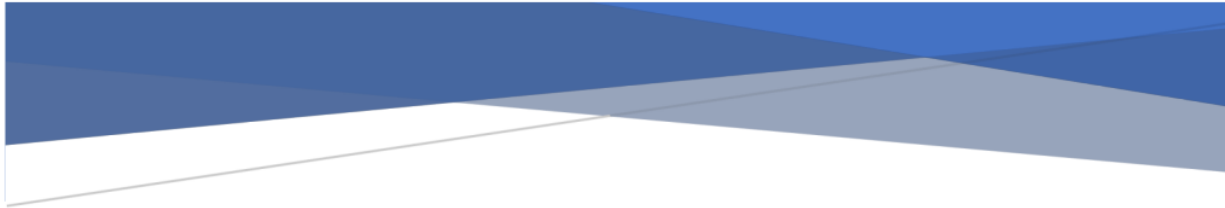
Submission date: 10-Dec-2021 11:57PM (UTC+0000)

Submission ID: 165852472

File name: 31344_Daniel_James_Bennett_Assignment_1_382340_782742866.docx (112.73K)

Word count: 1136

Character count: 6186



ASSINGMENT 1

Comparing SOAP, REST and GraphQL

Daniel Bennett
191798

Contents

Introduction 2

SOAP 2

 Benefits..... 2

 Drawbacks..... 2

REST..... 2

 Benefits..... 3

 Drawbacks..... 3

GraphQL 3

 Benefits..... 3

 Drawbacks..... 3

Conclusion 4

Link to YouTube Video 4

References 4

Introduction

In this report I will be conducting an analysis of three different web services and debating the principal benefits and drawbacks of each of them. By the end of the document, it is my intention to have a clear understanding of these and to be able to create a short conclusion on my findings.

SOAP

SOAP stands for Simple Object Access Protocol. Originally developed by Microsoft, IBM and others, it was intended to help bridge the gap between the many devices that communicated via the Internet. The protocol was developed to easily move XML packages across HTTP.

To achieve this SOAP on the client side would wrap messages, this would allow the SOAP listener on the server to decode this message and interact with the original wrapped up script/request.[1]

Benefits

One large benefit of SOAP is that it does not require a strong connection between client and server, this means that it has the ability to be more useful across more areas. This allows for longer periods of inactivity before the request would timeout. [2]

SOAP also is able to act as a compatibility layer between programs written in different languages which otherwise would cause issues. As the message wrapped in the SOAP protocol, it has to be decoded at the other end using SOAP.

Another of the main benefits of SOAP is its security, there are many methods to secure the envelopes the SOAP transmits with. And due to the large payload already, the overhead of more encryption and security does not a huge detriment on performance than normal.

Drawbacks

One of SOAP's larger drawbacks is it's required overhead. It required additional XML processing, which ends up being slower when compared to its competitors simply due to the payload being very large, even for a simple string message.

In addition to this, SOAP does not use other file types, it is exclusive to XML. This means that you cannot use JSON for example. [3]

REST

REST is an acronym for Representational State Transfer. It was first presented in 2000 by Roy Fielding [4]. REST was built to be simple, lightweight, and fast. Using key architectural design traits and constraints, this was made to enable services to work as best as they can on the web. [5]

Benefits

Due to REST's design constraints, it has many benefits. One key factor of that design is a "uniform interface". This means that when used in a web service, it will have desirable properties. Such as, "performance scalability and modifiability" [5].

Again, due to the design of REST it is all resources can be interacted with simple, well-defined operations. This leads to API's being incredibly easy to develop and required fewer resources to run.

Drawbacks

As part of REST's architecture design, it is stateless, meaning that all communication between the client and server, has all the information required to complete the request. This means that it doesn't do well in situations where a state is required to be maintained on either the client or server. Due to this, in scenarios that require heavy processing, this ends up being left on the client. For example, if the client is using a shopping website, and they are storing items in their cart, this is required to be maintained by the client. In certain scenarios this could lead to the client becoming slow due to the processing required.

One of the main drawbacks to REST is its lack of security, due to its stateless design it leaves is a risk in itself as it transports all information in a clear state. However, this does make it extremely useful for public information such as public APIs. [6]

GraphQL

GraphQL is a query language for APIs and runtime. This means it provides a complete package for all your data transfer needs. GraphQL is built on the concept of asking for what you need and getting exactly that. It also allows for many queries to be completed simultaneously in a single request. GraphQL API design uses single endpoints for communication allowing you to access the full capabilities of your data.[7]

Benefits

GraphQLs primary benefit is the ability to perform all communication at a single endpoint. This means all your data communication can be completed with a single query between client and server. This means that you only receive the data that you actually need and you do not end up with missing data due to a single faulty endpoint. [8]

Drawbacks

Due to bundling everything into a single query, when there is a need to make lots of small exchanges it is beat out by competitors that are built on the concept of doing this, for example REST.

In addition to this, GraphQL does not have caching in exchange for efficiency and speed. Therefore, if this is necessary for you then it will have to be made by you. [9]

Conclusion

In conclusion, there are obvious use cases for each service. SOAP although seeming the worst out of all of them, is by far the most secure service. It is also the most reliable due to its long timeout periods.

REST is an easy middle ground and has been used by lots of people, therefore, there is lots of varieties usable for it. However, even though it is very good, GraphQL seems to cover most of what REST is good at and does it better.

I think GraphQL will end up replacing REST in the next few years, but like SOAP, it will always have a use and I am sure a lot of people will keep using it.

Link to YouTube Video

https://www.youtube.com/watch?v=ECn3_Op437k

References

- [1]m. Mehta, "Introduction to SOAP", *C-sharpcorner.com*, 2021. [Online]. Available: <https://www.c-sharpcorner.com/article/introduction-to-soap/>. [Accessed: 10- Dec- 2021]
- [2]"SOAP web service", *Docs.servicenow.com*, 2021. [Online]. Available: https://docs.servicenow.com/bundle/rome-application-development/page/integrate/inbound-soap/concept/c_SOAPWebService.html. [Accessed: 10- Dec- 2021]
- [3]"What are the disadvantages of SOAP web services? - ...", *Onlineinterviewquestions.com*, 2021. [Online]. Available: <https://www.onlineinterviewquestions.com/what-are-the-disadvantages-of-soap-web-s/>. [Accessed: 10- Dec- 2021]
- [4]D. Young, J. Packer, C. Choudhary, R. Talada, P. Metkar and S. Masne, "What is REST", *REST API Tutorial*, 2021. [Online]. Available: <https://restfulapi.net>. [Accessed: 10- Dec- 2021]
- [5]"What Are RESTful Web Services? - The Java EE 6 Tutorial", *Docs.oracle.com*, 2021. [Online]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>. [Accessed: 10- Dec- 2021]
- [6]"Advantages and Disadvantages of REST API", *Krify - Web and Mobile App Design & Development Company in India & UK*, 2021. [Online]. Available: <https://krify.co/advantages-and-disadvantages-of-rest-api/>. [Accessed: 10- Dec- 2021]
- [7]"GraphQL | A query language for your API", *Graphql.org*, 2021. [Online]. Available: <https://graphql.org>. [Accessed: 10- Dec- 2021]

- [8]"GraphQL vs REST - A comparison", *Howtographql.com*, 2021. [Online]. Available: <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>. [Accessed: 10- Dec- 2021]
- [9]"GraphQL VS REST – Benefits and Code Example Comparisons", *freeCodeCamp.org*, 2021. [Online]. Available: <https://www.freecodecamp.org/news/graphql-vs-rest-benefits-and-code-example-comparisons/>. [Accessed: 10- Dec- 2021]

Assignment 1

GRADEMARK REPORT

FINAL GRADE

95 / 100

GENERAL COMMENTS

Instructor

Ajax & REST (45): 43

Written Research (45): 42

Demo (10): 10

Feedback:
=====

Quite like this report, would like to have had characteristics of the services identified for comparison.

Movie site.

In some places, you use a POST request to read data. In REST, we always use POST to ADD new data to the resource.

Uses Axios from web service you created to call through a film database service. Nice.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6
