

Investingating SOAP, REST & GraphQL

by Richard Lamnea

Submission date: 16-Dec-2021 07:39PM (UTC+0000)

Submission ID: 166668056

File name: 11200_Richard_Lamnea_Investingating_SOAP_REST_GraphQL_382340_314088055.docx
(124.41K)

Word count: 1665

Character count: 10200

University of Wales Trinity St David

School of Applied Computing

Investigating SOAP, REST and GraphQL web services

Richard Lamnea

ACCA6024



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

Investigating SOAP, REST and GraphQL web services.

Richard Lamnea

School of Applied Computing, University of Wales Trinity St David, Swansea, Wales

1802749@student.uwtsd.ac.uk

TABLE OF CONTENTS

INTRODUCTION	1
I. SOAP WEB SERVICE	1
A. Overview	1
B. Benefits	1
C. Drawbacks	1
II. REST WEB SERVICE	1
A. Overview	2
B. Benefits	2
C. Drawbacks	2
III. GRAPHQL WEB SERVICE	2
A. Overview	2
B. Benefits	3
C. Drawbacks	3
IV. IMPLEMENTATION OF REST FRONTEND APPLICATION	3
V. CONCLUSION	3
VI. REFERENCES	3

INTRODUCTION

This report discusses an overview of SOAP, REST and GraphQL web services. Also outlined are the benefits and drawbacks of these services.

I. SOAP WEB SERVICE

Simple Object Access Protocol (SOAP) is a messaging protocol for web services to exchange data via HTTP and XML.

A. Overview

SOAP utilises XML (Extensible Markup Language) as its messaging format to send and receive messages and depends on application layer protocols such as HTTP and SMTP. Three roles define the SOAP service architecture, the service provider,

the service requester and the service registry [1]. Each component can communicate with the other via three operations [1]. These operations are Publish, Find and Bind [1]. A service provider can publish a service description to a service registry. A Service requester can find a service description from the service registry [1]. A service requester can bind a service description with a service provider [1]. The bind operation allows the service provider to interact with the web service implementation.

The SOAP message is structured to contain two sub-elements [2]. The message itself is called a SOAP Envelope and contains processing instructions for the receiver [3]. The envelope includes an optional SOAP header element and a mandatory SOAP body element [2]. The header contains application information, such as authentication details. The body element has the data intended for the recipient and a fault element for error handling [4].

B. Benefits

SOAP is characteristically neutral, allowing the service to be used with other transport protocols. The majority of SOAP implementations utilise HTTP protocol. However, SOAP can also be used with SMTP, FTP and JMS [5], [6].

SOAP provides built-in error Handling within the envelopes' body element. If a problem arises with the request, a response will provide error information [7].

Extensions to the SOAP protocol can be added to improve its service. WS-Security can be included within the envelope and incorporate security features such as encryption, security tokens, certificates and authentication details [8]. These extensions to the protocol improve data security and interoperability [8].

C. Drawbacks

SOAP web service was deprecated in 2011, meaning it will not have new features or updates. However, extensions are still available and updated [9].

SOAP utilises XML format for its messaging system due to the large payload size within each envelope, causing the protocol to be considered slow [3].

II. REST WEB SERVICE

REST (Representation State Transfer) is an architecture approach for implementing an API or web service [10]. If the

service follows the principles and constraints explained by the architecture, then the service can be referred to as RESTful [11].

A. Overview

The REST architecture comprises six guiding principles for implementing a REST API:

- 1) Uniform Interface
When creating an interface for communication between a client and a service, the interface must adhere to 4 constraints [11].
 - Identification of resources [11].
 - Manipulation of resources through representations [11].
 - Self-descriptive messages [11].
 - Hypermedia as the engine of application state [11].
- 2) Client-server
The REST API must follow a client-server design pattern with separation of concerns in mind[12] [11]. This allows loose coupling of the client and server when developing components [10].
- 3) Stateless
This principle mandates that every request from the client to the server must contain all required information to complete a request, and the client must retain the session state[11].
- 4) Cacheable
This constraint requires that a response be labelled as cacheable or not [11].
- 5) Layered system
The layered system constraint is designed to improve behaviour scale requirements. Each component cannot see beyond the interface they communicate with [11].
- 6) Code on demand
REST also allows the function to be extended to enable downloading and execution of code to ease client load [11].

REST outlines a set of data elements. These are:

- Resource.
This is the object that can be retrieved.
- Resource identifier
This is the address where the object can be found. E.g., URL.
- Representation
This is the format of the returned object. E.g., HTML document.

Additional metadata elements include the resource media type, the resource source link, and cache control [11].

The REST architecture defines identifiers for each resource when a client and server interact. The request & response should specify the content type and resource method. REST has five resource methods which are:

- HTTP GET
This method requests a resource for retrieval [11].
- HTTP POST
This method allows a resource to be manipulated[11].
- HTTP PUT
This method is used to create a new resource[11].
- HTTP DELETE
This method is used to delete a resource[11].
- HTTP PATCH
This method is used to manipulate a resource partially [11].

B. Benefits

The Benefits of utilising a RESTful web service are:

Messages between client and server are created in a simple format. RESTful web services are relatively cheap and easy to learn and implement [13].

REST supports stateless communication [13]. Also, as messages are sent in a JSON format, most clients can consume the data quickly, meaning that the bandwidth required is small [13].

C. Drawbacks

There are also disadvantages to using REST within a web service, such as:

REST lacks support for standards which include security & policy standards[13].

Due to the client-server design pattern, REST web services struggle to handle large amounts of data. Processing large sums of these requests can cause latency and delays[13].

III. GRAPHQL WEB SERVICE

GraphQL is a query language and runtime explicitly designed for a web service (API). The GraphQL API allows a client to request precisely the data they require [14].

A. Overview

When a GraphQL web service is constructed, fields and their types are defined in a schema and functions are created for each field. This data is exposed as a graph containing nodes representing objects with fields [15]. GraphQL utilises three resource methods which are:

- Query.
Read the data[16].
- Mutation.
Write or mutate the data[16].
- Subscription.
Continuously read the data[16].

GraphQL follows three main principles and multiple sub principles when implementing a schema. The three main principles are:

- Integrity Principle
A schema should utilise a stable and defined graph even across multiple teams[17].
- Agility Principle
Approach development with a continuous abstract approach and repeatedly develop incrementally [17].
- Operations Principle
Utilise a layered architecture, and implement structured and secure access to the graph on a per-client basis[17].

B. Benefits

The benefits of utilising a GraphQL web service are:

When a request is sent via GraphQL, the client specifies the data they require, allowing a precise and performant procedure[18].

Implementation of GraphQL can be complex; however, once the schema has been defined, repetitive logic programming found in REST is no longer needed[19].

C. Drawbacks

The drawbacks of using a GraphQL service are:

As GraphQL utilises its own strongly typed language, a developer must learn the Query language causing more complexity to the initial implementation [20].

IV. IMPLEMENTATION OF A FRONTEND REST APPLICATION

An implementation of a REST frontend application was created for demonstration which can be viewed at:

<https://www.youtube.com/watch?v=eqZj1eR01sc>

V. CONCLUSION

In conclusion, An implementation of a REST API over SOAP or GraphQL is the easiest and most cost-effective solution. However, if the data required for the application is extensive, then the utilisation of a GraphQL API would provide significant benefits. If the data necessary for the application needs to be securely transmitted, then a SOAP implementation is a viable contender.

VI. REFERENCES

- [1] 'SOAP web services architecture - IBM Docs', Dec. 03, 2021. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/was/9.0.5?topic=services-soap-over-jms-protocol> (accessed Dec. 06, 2021).
- [2] 'Structure of a SOAP message - IBM Docs', Jun. 22, 2021. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/cics-ts/5.4?topic=format-structure-soap-message> (accessed Dec. 05, 2021).
- [3] M. Ali, M. F. Zolkipli, J. M. Zain, and S. Anwar, 'Mobile Cloud Computing with SOAP and REST Web Services', *J. Phys.: Conf. Ser.*, vol. 1018, p. 012005, May 2018, doi: 10.1088/1742-6596/1018/1/012005.
- [4] 'SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)'. <https://www.w3.org/TR/soap12-part1/Overview.html> (accessed Dec. 06, 2021).
- [5] H. Sugawara and S. Miyazaki, 'Biological SOAP servers and web services provided by the public sequence data bank', *Nucleic Acids Research*, vol. 31, no. 13, pp. 3836–3839, Jul. 2003, doi: 10.1093/nar/gkg558.
- [6] 'SOAP over JSM protocol - IBM Docs', Oct. 13, 2021. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/cics-ts/5.3?topic=format-soap-web-services-architecture> (accessed Dec. 06, 2021).
- [7] 'SOAP Error Handling'. https://ftpdocs.broadcom.com/cadocs/0/CA%20Service%20Desk%20Manager%2012%207-ENU/Bookshelf_Files/HTML/CA_SDM_Impl_ENU/soap_error_handling.html (accessed Dec. 06, 2021).
- [8] 'Basic Security Profile - Version 1.0 (BdAD)'. <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html> (accessed Dec. 06, 2021).
- [9] M. S. Warvante, 'A Survey on Developing Web Services with SOAP and REST', vol. 6, no. 10, p. 7.
- [10] 'What is REST', *REST API Tutorial*. <https://restfulapi.net/> (accessed Dec. 10, 2021).
- [11] 'Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)'. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (accessed Dec. 10, 2021).
- [12] 'Principles of Software Engineering'. <https://www.d.umn.edu/~gshute/softeng/principles.html> (accessed Dec. 10, 2021).
- [13] Institute of Electrical and Electronics Engineers, Ed., *2012 International Conference on Information Technology and e-Services (ICITeS 2012): Sousse, Tunisia, 24 - 26 March 2012*. Piscataway, NJ: IEEE, 2012.
- [14] 'GraphQL | A query language for your API'. <https://graphql.org/> (accessed Dec. 10, 2021).
- [15] G. Brito and M. T. Valente, 'REST vs GraphQL: A Controlled Experiment', *arXiv:2003.04761 [cs]*, Mar. 2020, Accessed: Dec. 11, 2021. [Online]. Available: <http://arxiv.org/abs/2003.04761>
- [16] T. Andersson and H. Reinholdsson, 'REST API vs GraphQL - A literature and experimental study', p. 43.

- [17] 'Principled GraphQL', *Principled GraphQL*.
<https://principledgraphql.com/> (accessed Dec. 14, 2021).
- [18] '4 reasons why you should use GraphQL over REST APIs',
DEV Community. <https://dev.to/blessingartcreator/stop-using-rest-for-apis-53n> (accessed Dec. 14, 2021).
- [19] 'Main advantages of GraphQL as an alternative to REST',
Nov. 27, 2018.
<https://www.adaltas.com/en/2018/11/27/graphql-advantages-over-rest/> (accessed Dec. 14, 2021).
- [20] 'Advantages and Disadvantages of GraphQL | Stable Kernel'.
<https://stablekernel.com/article/advantages-and-disadvantages-of-graphql/> (accessed Dec. 14, 2021).

Investingating SOAP, REST & GraphQL

GRADEMARK REPORT

FINAL GRADE

96/100

GENERAL COMMENTS

Instructor

Ajax & REST (45): 45

Written Research (45): 41

Demo (10): 10

Feedback:
=====

The report could have expanded on SOAP.

Movie website, looks very nice and is a very detailed application. Excellent.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5