

Comparing SOAP, REST and GraphQL web services

Comparing SOAP, REST and GraphQL web services

- What is SOAP?

 - How does it work?

 - Advantages of SOAP

 - Disadvantages of SOAP

- What is REST?

 - Advantages of REST

 - Disadvantages of REST

- What is GraphQL?

 - How does it work?

 - Advantages of GraphQL

 - Disadvantages of GraphQL

- Resources

Demonstration Video: <https://youtu.be/FZhu6mE3pR0>

What is SOAP?

SOAP (Simple Object Access Protocol) is a standard of sending messages defined by the world wide web consortium. All SOAP based messages are based on the XML format, which means SOAP relies on XML when sending and receiving messages. SOAP is independent from a specific programming language, which makes it extremely flexible. This means that any language can be used to both send and receive a SOAP message, for example, a SOAP message can be sent via a program written in Python, and received by a program written in C#. SOAP is implemented in most modern programming languages, making it extremely versatile. [1*]

SOAP has built in error handling, which means that if there's a problem with a SOAP request, the SOAP API can return an error message outlining the issue with that particular request. Developers can use the information from error message to fix issues with requests. This is extremely helpful for developers, especially if the web service isn't owned by the developer in question. With error handling, developers aren't totally left in the dark when a request doesn't work.

SOAP doesn't need to be sent using the HTTP protocol either, SOAP requests aren't limited to one protocol, which means SOAP messages can be sent and received through multiple protocol, which makes it that much more versatile, especially for developers who aren't specifically developing a website. [2*]

How does it work?

The standard SOAP API request message and response message consists of four main elements enveloped using tags:

1. **The header (optional)**

This is where the additional specific parts of the message go, such as authentication information etc.

2. **The body**

This is where the actual request or response of the message is stored.

3. **The fault (optional)**

This is where data about any errors that could have emerged from the request or response are stored.

SOAP is sent via a variety of transfer protocols, both on the high and low level. SOAP allows messaging via such protocols as:

- **TCP (Transaction Control Protocol)**

Commonly used protocol that enables applications and computing devices to exchange messages over the internet.

- **SMTP (Simple Mail Transfer Protocol)**

A protocol responsible for sending email messages from one account to another via an internet connection.

- **FTP (File Transfer Protocol)**

A protocol responsible for transferring, uploading and downloading files between computer systems through the internet.

And any other text based transfer protocol

However, it doesn't make much sense to use SOAP with many protocols other than HTTP(S) as SOAP was primarily designed to work with HTTP(S). However, its versatility is appreciated for niche scenarios that may benefit from its flexibility, such as security constraints or server requirements [3*]

Advantages of SOAP

One advantage of SOAP is that it's standardised. SOAP is standardised by the World Wide Web Consortium (W3C) which makes SOAP a protocol that is universally accepted, which makes SOAP easily accessible and available for more open source implementations.

SOAP also makes it easier operate across firewalls and proxies due to the standard HTTP protocol that SOAP uses, this means developers don't need to consider issues with firewalls, as most firewalls allow connections from the HTTP protocol, the protocol responsible for web services.

Despite often pairing SOAP with the HTTP protocol, in rare cases, unlike REST, SOAP can be used with protocols other than HTTP. In rare cases developers require better transactional reliability, in which case SOAP would be a more versatile alternative to REST, which needs to be sent via the HTTP protocol [4*].

Disadvantages of SOAP

SOAP messages are defined in a human readable language (much like HTML), known as Web Services Definition Language (WSDL) which is written in XML format, a format similar to HTML that uses tags [4] to define data. This makes it fairly easy for us humans to understand and implement the data, however, XML being used as the standard language means that SOAP messages are larger than most alternatives, such as Common Object Request Broker Architecture (CORBA) which is significantly smaller, making SOAP a slow method for microservice-based web services, commonly used today [5*]

What is REST?

REST, which stands for 'Representational State Transfer' is an API protocol, much like SOAP. However, if a program requests a resource from a server using the REST architecture, the server will typically respond with the current state of the requested resource, but this time the data is represented in a standardised format, as opposed to a block of XML. The format the data is sent via is decided by the owner of the API. Unlike SOAP, developers can also integrate the ability to edit, add or delete data via REST requests. This means that with a REST API, not only can clients receive data from your web service, they can add data and edit data off-site as well. (Depending on permissions). [6]

Advantages of REST

Flexibility is one of REST's major advantage compared to SOAP, unlike SOAP, REST is not constrained to XML, which means requests from a REST API can be retrieved in XML, JSON, YAML or any other language the client requests the data in. [7]

Disadvantages of REST

In terms of security, REST isn't as secure as SOAP, which is why REST is often used as a public API. If you need to transfer secure private data from the client to the server, REST wouldn't be an appropriate API method to use [8]

What is GraphQL?

GraphQL is a relatively new open source API standard developed by Facebook in 2012 and was made open source in 2015 and is since maintained by the GraphQL Foundation, which is currently hosted by the Linux Foundation. [9]

GraphQL prioritises giving clients the exact data that was requested whilst also providing flexibility and fast performance, by retrieving all the data you need in a single request, as opposed to REST in which loading from multiple URLs is often required. [10]

How does it work?

GraphQL uses Schema Definition Language [11]

A schema is much like a middle ground manifesto that defines what the API can and cannot do, including what data the clients can and cannot request or change. This is what makes GraphQL flexible, as the schema details what the client can and cannot do with the API without exposing the back-end implementations.

The GraphQL schema also allows setting descriptions to both types and fields in the form of a comment, which makes documentation easier for both the client and the developer.

For example:

```
1  """
2  Information about the student
3  """
4
5  type uniStudent
6  {
7      "The age of the student"
8      age: Int
9
10     "The student ID"
11     id: String
12
13     "The current assignment of the student"
14     currentAssignment: [Assignment]
15 }
```

[12]

Advantages of GraphQL

If, for example, a client requested data from a server about a specific university student, but only required their Student name and student ID, if using a REST API to retrieve that data from a server, the REST API will retrieve all the data on that particular student, which can be performance heavy and unnecessary.

With GraphQL, due to the flexibility of the API, a client can request not only data for the student, but specific data from the server such as their Student ID and name. This flexibility is an advantage over REST, as only the requested data is retrieved from the server, which heavily impacts performance on a positive scale, as this method

actively prevents ‘over-fetching’, which is when the client downloads more information than is actually necessary (in this example, the client would receive all data about this particular student, despite only needing the student name and ID). [13]

Disadvantages of GraphQL

One major disadvantage of GraphQL is rate limiting. With a REST based API, you can simply limit the amount of daily requests by specifying a daily request limit. This is a bit more difficult with GraphQL due to the way requests with GraphQL are made. [14]

Implementing cache control with GraphQL is especially difficult compared to other protocols such as REST, this is because GraphQL queries are often different from each other, which in turn makes caching incredibly difficult to implement, even when operating on the same entity. Luckily, many libraries built on-top of GraphQL offer solutions such as efficient caching mechanisms, but compared to REST’s ability to cache resources based on the URL, caching is one of GraphQL’s disadvantages. [15]

Resources

- [1] ‘SOAP API | SOAP API Example Protocol | SOAP APIs Interface’, *stoplight.io*. <http://stoplight.io> (accessed Dec. 09, 2021).
- [2] ‘SOAP vs REST. What’, *SmartBear.com*. <https://smartbear.com/blog/soap-vs-rest-whats-the-difference/> (accessed Dec. 09, 2021).
- [3] ‘What is SOAP: Formats, Protocols, Message Structure, and How SOAP is Different from REST’, *AltexSoft*. <https://www.altexsoft.com/blog/engineering/what-is-soap-formats-protocols-message-structure-and-how-soap-is-different-from-rest/> (accessed Dec. 09, 2021).
- [4] ‘SOAP vs. REST: The Differences and Benefits Between the Two Widely-Used Web Service Communication Protocols’, *Stackify*, Mar. 14, 2017. <https://stackify.com/soap-vs-rest/> (accessed Dec. 09, 2021).
- [5] ‘What is SOAP (Simple Object Access Protocol)? - Definition from WhatIs.com’, *SearchAppArchitecture*. <https://searchapparchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol> (accessed Dec. 09, 2021).
- [6] ‘What is a RESTful API?’, *MuleSoft*. <https://www.mulesoft.com/resources/api/restful-api> (accessed Dec. 09, 2021).
- [7] J. Juviler, ‘REST APIs: How They Work and What You Need to Know’. <https://blog.hubspot.com/website/what-is-rest-api> (accessed Dec. 09, 2021).
- [8] ‘IBM Docs’. https://www.ibm.com/docs/en/radfwf/9.6?topic=SSRTLW_9.6.0/org.eclipse.jst.ws.doc.user/concepts/cwsdl.html (accessed Dec. 09, 2021).

- [9] 'GraphQL | A query language for your API'. <https://graphql.org/> (accessed Dec. 09, 2021).
- [11] 'What is the GraphQL Foundation? | GraphQL'. <https://graphql.org/foundation/> (accessed Dec. 09, 2021).
- [12] 'Schema definition language (SDL) | Lift-off I: Basics | Apollo Odyssey', *Schema definition language (SDL)*. <https://odyssey.apollographql.com/lift-off-part1/schema-definition-language-sdl/> (accessed Dec. 09, 2021).
- [13] 'GraphQL Advantages and Disadvantages - javatpoint', [www.javatpoint.com. http://www.javatpoint.com/graphql-advantages-and-disadvantages](http://www.javatpoint.com/graphql-advantages-and-disadvantages) (accessed Dec. 09, 2021).
- [14] 'GraphQL: Core Features, Architecture, Pros and Cons', Mar. 23, 2019. <https://www.altexsoft.com/blog/engineering/graphql-core-features-architecture-pros-and-cons/> (accessed Dec. 09, 2021).
- [15] 'GraphQL vs REST - A comparison'. <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/> (accessed Dec. 09, 2021).