University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Review of Web Service Technologies: REST over SOAP

Sarah Hussein Toman

Roads and Transport Department, College of Engineering, University of Al-Qadisiyah, Sarah.toman@qu.edu.iq

ARTICLE INFO

Article history:
Received: 15/11/2020
Rrevised form: //
Accepted : 22/11/2020
Available online: 25 /11/2020

Keywords:

Web Services, RESTfull Services

SOAP based Services, SOAP API

REST API

ABSTRACT

Recently, most of the web applications depend on the web services to solve the application integration problem. Web service is an intermediate software that allows the interaction between the applications written by different languages. Web Service is a client-server, modular, self-contained, and dynamic application that ensures the communication between the applications via the World Wide Web (WWW). The aim of this paper is to choice right kind of Web services at design stage of the project. Thus, we provide an overview of web services. Next, we clarify the architecture of web services. After that, we focus on the most popular kinds of web services, the Representational State Transfer (REST) and the Simple Object Access Protocol(SOAP) then explain APIs details. Finally, we address the performance evaluation of the previous works and explained the results as comparison between SOAP and REST. A recommendation to use RESTful web services has been concluded from experimental result.

MSC : 30C45 , 30C50

DOI : https://doi.org/10.29304/jqcm.2020.12.4.715

1. Introduction

Nowadays is the ear of the internet, the user communicates to applications, furthermore, there is application to application interaction[1,2]. Web Service is the most common way for applications to applications communicate across an internet protocol backbone [1,3]. The World Wide Web Consortium(W3C) provided a more price definition to Web Services as: “a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols” [4].

⁎***Corresponding author :*** *Sarah Hussein Toman*
Email addresses: Sarah.toman@qu.edu.iq
Communicated by : Alaa Hussein Hamadi

Unlike Web application, Web services do not provide the user GUI, rather than it designed to exchange data among different languages and platform-based applications through a programmatic interface over the network called a WEB API [5].

SOAP and REST are Web services' types. SOAP stands for Simple Object Access Protocol. SOAP, XML-messaging protocol that transfers the data over HTTP protocol. SOAP have restrictions over the representation format, is accepted only XML format [6]. Representational State Transfer(REST) is an architectural style for web services. In REST, the server responds to client requests by using any resources allowed in computer systems[7]. The representation format can be in JSON or XML. Both Web Services types enable heterogeneous environments and languages based applications to communicate with each other [8,9].

The purpose from this paper is to choice which of Web services technologies that should be use at design stage of the project. To achieve this purpose, we explain APIs details, address the performance evaluation of the previous works and clarified the results as comparison between SOAP and REST.

2. Related Work

K. Wagh & R. Thool (2012) listed out the comparison between the REST and SOAP frameworks and found the REST is more suitable for the wireless communication than SOAP. In another hand, the SOAP is more secure than REST for the same purpose [10]. S. Mumbaikar & P. Padiya (2013) demonstrated the REST and SOAP-based web services comparison for the mobile environment depend on the different metrics and concluded that the RESTfull web services is better than the SOAP-based web services in the performance, where The RESTful web services are Self-descriptive, lightweight, easy with high flexibility and lower overhead [1].

M. Massé (2013) presented the set of rules to design the REST API then addressed the needs of clients and why choose the REST API [11]. P. Giessler & et.al (2015) categorize the best 23 practices for developing RESTfull services into 8 classes and explain the practice of web services on diverse applications [12].

D. Qiu & et al. (2016), described the development of API in Java programming language and analyzed about 5000 Java projects(open-source projects). They reached from this research that the deprecated APIs are broadly used more than the core APIs [13]. J. Tihomirovs & J.

Grabis (2016) compared between the REST and SOAP APIs by using the software evaluation metrics and found out the REST is lower than SOAP in response time. Further, the SOAP is less than REST in the number of line code [14].

F. Halili & E. Ramadani (2018) clarified the advantages and disadvantages of RESTfull web services and SOAP-based web services, in addition to a comparison study between them [15]. A. Soni et al (2019) presented the comparative analysis between REST API and SOAP API based on response time, execution speed, and memory usage features [16].

3. Background Study

3.1 Web Service Architecture

Service provider, service requester, and service registry are composed the Web Service architecture. Figure (1) exhibits the visual view of Web Services architecture. Service provider creates the Web services and provides the implementations and interface of it after that publishing the service descriptions in the service registry. Service Registry catalogs the information of Service descriptions that contain the details of service API, implementation and the location (where the service is found), and looking for the desired service when the Service Requester sends a query. Then, the Service Requester can interact with Service Provider [17].

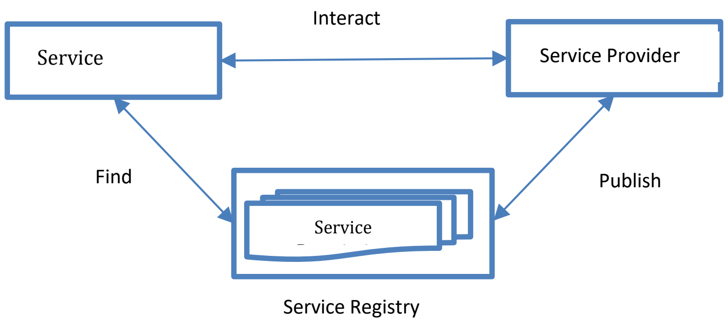


Figure 1- An architecture of Web services

3.2 Web Services Characteristics

Web services have many characteristics are listed below [9]:

1. XML-Based – Web services depend upon XML for data transportation and representation. The use of the XML prevents the binding from occurs in the network or the platform or the operating system.

2. Loose coupling – the web service client and the server are not tied closely. So, there is no direct link between them. By adopting an architecture –based loosely coupled tends to facilitate the management of the software systems and the integration among them.

3. Adaptability – the web services can work by using a synchronous or asynchronous method. the synchronous means the service consumer cannot execute any operation until receiving the response from the server. while asynchronous allows the parallel execution without waiting for the response.

4. Reusability – the web services are available on the internet, therefore, it provides the meaning of code reusable.

5. Interoperability – web services can be sharing the data between varied applications (i.e written by different languages).

6. Automatic Discovery - Web Services allow the client to easily retrieve the web service from the Service Providers by sending the query to the service registry.

3.2 SOAP API

Simple Object Access Protocol, the standard messaging protocol that enables computing systems that are built on different kinds of programming languages to communicate with each other by used XML messages over HTTP protocol. SOAP Protocol follows distributed systems in some features such as security, privacy, the quality of attachment, and reliability [15]. SOAP is an independent platform, hence, it can work with any programming language and operating system. The main concept of SOAP is the Provider publishes the service in the registry then the requester binds it according to registry query answer [18]. The request and the respond calls appear as SOAP messages, Figure.2 represents the structure of the SOAP message and figure.3 display a sample of SOAP message. the Web Service Description Language(WSDL) defines the XML message structure [19]. The structure of SOAP message consists of the following elements [20]:

- **The Envelope element**: it's the root element that identify and encapsulate the XML document's details in the SOAP message.
- **The header element**: this element contains optional information related to specifics or requirements of documents such as the authentication.
- **The body element**: it's a necessary element, that contains request and response information that directed between web service and application.
- **The Fault element** (Optional): it shows error information that occurs during request and response message, as well, status information.

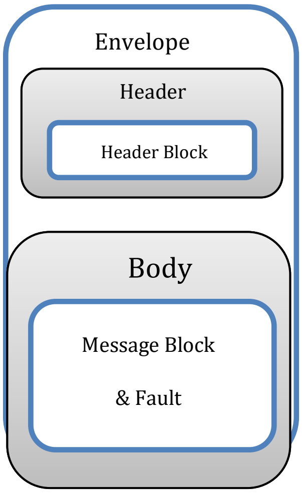
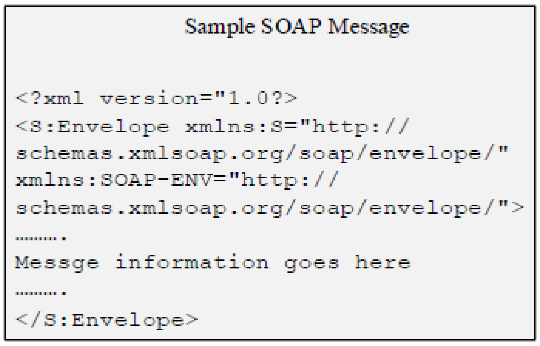


Figure.2 Structure of SOAP Message



Sample SOAP Message

```
<?xml version="1.0?>
<S:Envelope xmlns:S="http://
schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENV="http://
schemas.xmlsoap.org/soap/envelope/">
……….
Messge information goes here
……….
</S:Envelope>
```

Figure.3 Sample of SOAP Message

3.3 REST API

Representational State Transfer (REST) is an architecture style that has a set of constraints and rules to determine Application Programming Interface(API) which allows each application to talk with others. REST follows HTTP methods to perform the communication operations (request and response) then return the response in JSON/XML format to the client [21]. Roy Fielding (2000) defined REST constraints in his dissertation in 2000. These constraints are listed below:

- **Client-Server:** this constrain aims to improve the flexibility of the user interface by separate the data storage from the user interface. Also, simplify the server components will improve the scalability. Hence, separation of concerns is an essential aspect behind the Client-Server constraint.
- **Stateless**: the principle of this constrain depend on no recourses management, in other words, the server will not have information about the previous request, it replies

according to the information found in the current request. The stateless communication between the client and the server enhanced reliability, visibility, and scalability.

Cache: as we mentioned in stateless constrain which make the nature of the request is independent and to save networking traffic between the client and server, the cache resides on the client to store the responds data then reuse them for equivalent requests if the response is cacheable. Cache constrain enhancing scalability, efficiency and reduces the average latency of some interactions.

- **Uniform Interface:** REST-style has four constraints to design a Uniform Interface that improved and simplified the interactions between the components. the REST uniform interface represented by identifying the resources, manipulation of the resources by the representation, self-description based messages, and hypermedia.
- **Layered System:** The layered system style prompts hierarchical layers based constraining component in an architecture, so each component cannot "see" the immediate layer interaction.
- **Code on demand:** REST Style allows to extend the client functionality by downloading and executing code, that code can be in the form of scripts or applets. This simplified pre-implemented at clients by reducing the required features.

3.3.1 REST Data Elements

1- **Resources:** any information that can be the destination of hypertext references. The information may be an image, a document, person, collection of resources, and so on[22, 23].

2- **Resources Identifier:** It identifies each resource that communicates between components by a Uniform Resource Identifier (URI) [22, 23].

3- **Resources Representation:** REST represents the current actions state of recourses by data and metadata to describe these data, then transfers it between components [22, 23].

3.3.2 Resource Methods

REST API develops over HTTP, hence, it follows HTTP methods to define the action for request and response to communication between the components. To perform the operations on the service, the following HTTP methods are used [22, 24]:

1- **GET**: Retrieve resource by request its URI.

2- **POST**: create/add new resource.

3- **PUT**: update the determined recourse.

4- **DELETE**: delete the determined recourse.

4. The Performance Evaluation

Many authors worked on the different metric to display the performance of SOAP and RESTful web services, here, we will display some of these works then extract the comparison on performance of SOAP and RESTfull web services based on different metrics:

In [25], The evaluation of a RESTful and SOAP web service on mobile devices implemented. Two benchmarks with the float and the string data types are used as parameters to measure a message size and a session time of service call. Table1. Shows the high performance RESTful over SOAP in a message size and a session time.

Table 1. Performance Comparison of Web Services mobile services [25]

Number of Array Elements	Message Size(byte)				Time(Milliseconds)			
	SOAP/HTTP		REST(HTTP)		SOAP/HTTP		REST(HTTP)	
	String Concatenation	Float Number Addition	String Concatenation	Float Number Addition	String concatenation	Float Number Addition	String Concatenation	Float Number Addition
2	351	357	39	32	781	781	359	359
3	371	383	48	36	828	781	344	407
4	395	409	63	35	828	922	359	375
5	418	435	76	39	969	1016	360	359
6	443	461	93	43	875	953	359	359

In table1, we noticed the average of message size of string concatenation are 395 and 63.8 for SOAP and REST respectively. Also, the average of message size of float number addition are 395 and 63.8 for SOAP and REST respectively. In addition, the time average of string concatenation was 856.2 and 371.8 for SOAP and REST respectively. Finaly, the time avarege of float number addition of SOAP and REST were 890.6 and 371.8, respectively. Hence, RESTful offers a good solution for the common of implementations, with lower overhead and higher flexibility.

On another hand, REST Web Services and SOAP Web services are tested for mobile computing by using Energy Consumption and Execution Time parameters in the matrix multiplication app (the developed prototype). Figure.4 shows the comparison between them. Conducting results display RESTfull is better than SOAP about 200% in the execution time and 250% in the energy consumption [26].

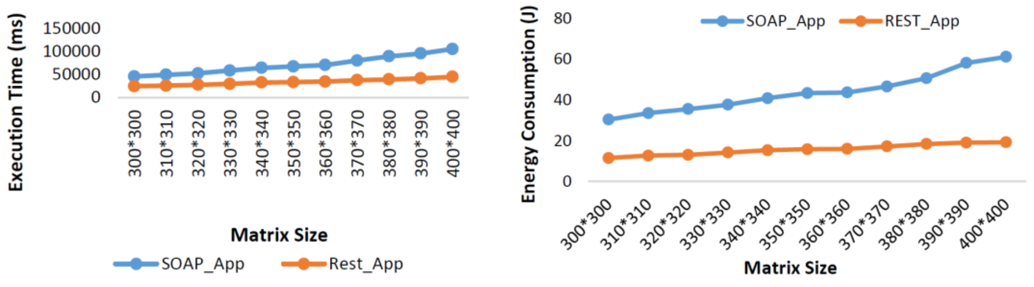


Figure 4. Performance Comparison of Web Services mobile services [26]

the WS-Security is one of the standards of SOAP protocol to encrypting messages so that the data transfer becomes safer, while, the REST approach is based on built in security HTTP (or HTTPS). the point-to-point communication protected by Transport layer security(TLS) but the problematic is happened if the mobile devices involved, TLS channel should be reset frequently [27]. Thus, SOAP based services is more Reliability than RESTfull services.

In [9] JAX-WS and JAX-RS are used to design SOAP API and REST API, respectively and tested by using a Postman (API testing tool) in terms of memory usage and response time. This gives insight view of which service is better to use as per requirements. The result of experiments demonstrations that the response time of REST is take about 4ms to 7ms less than

SOAP. It has been observed that as number of API increases, REST takes approximate 1MB to 2MB less memory usage than SOAP.

From the viewpoint of maintains, complexity, lines of code and cost, the development and implementation RESTfull is easier than the SOAP based services because the REST protocol is closely to the HTTP protocol and it uses basic operations of the HTTP for instance GET, POST, PUT, etc. In addition, it uses the standards of W3C / IETF (HTTP, XML, URI, and MIME) [14, 28].

4.1 Comparison Summary

By taking into consideration of the previously viewed. Table 2. Demonstrates the concluded results of the RESTfull and SOAP Web Services performance.

Table 2. Comparison of Web Services Performance (REST & SOAP)

Feature	**REST**	**SOAP**
Definition	Architectural Style	Messaging Protocol
	Representational State Transfer	Simple Object Access Protocol
	Follows A Set Of Constraints	Set Of Rules With A WSDL File That Contains The Location And The Other Information About The Web Service
	It Can Use SOAP Protocol	Cannot Use The REST Because Its Protocol
	Exposing About Its Components To The Clients By The Using URL	Exposing About Its Functionality To Clients By Using The Service Interfaces
Message Format	Json/Xml/Html	Xml
Response Time /Execution Time	Rest Permits Lower Bandwidth Usage	Soap Permits High Bandwidth Usage Because It Consumes More Resources And Based On Envelope Style
Memory	Low	High

Requirement/ Memory Consumption	(Parsing JSON That Used By REST Is Easier Than Parsing XML That Used By SOAP)	
Cost	Low	High
Client-Server Interaction	Loosely Coupled	Tightly Coupled
Payload	Heavy (Rest Uses Uri To Data Transfer)	Lightweight
Attachment Parsing	Any Data Types	Binary
Reliability	No	Yes
Environments	Point-To-Point Communication Model	Distributed Computing
Development	Simpler	Harder (SOAP Is Harder To Develop And Maintains Because It Needs More Requirements)
Line Of Code	More	Less
Errors Handling	More	Less
Security Complexity	High	Low
Performance	Slower	Faster
Server-Side Effort	Less	More
Client-Side Effort	More	Less
Server-Side Maintainability	Better	-
Client-Side Maintainability	-	Better
Verbose	Less	More

Tooling/Middleware Support	Less	More

5. Conclusion

As a result of the comparison between REST and SOAP services, we concluded that the RESTfull web services use all the features of HTTP, therefore, it is simple in the development and the maintains and. Furthermore, RESTfull web services easy to consume, fast, lightweight, self-descriptive, support all types of data, and need less (bandwidth, response time, and memory requirements). So, we can say, RESTfull web services will continue in the coming years to lead the more of the technology space.

References

[1] S. Mumbaikar and P. Padiya, "Web Services Based On SOAP and REST Principles", International Journal of Scientific and Research Publications, Volume 3, Issue 5, 2013.

[2] S.H. Toman , et al , Cluster-based information retrieval by using (K-means)- hierarchical parallel genetic algorithms approach, TELKOMNIKA Telecommunication, Computing, Electronics and Control Vol. 19, No. 1, February 2021, pp. 349~356

[3] R. F. Hassan , et al., "Improving the web indexing quality through a website-search engine coactions," International Journal of Computer and Information Technology, vol. 3, no. 2, March 2014.

[4] G. Alonso, et al, "Web Services: Concepts, Architectures and Applications", Springer, 2004.

[5] P. Adamczyk , et al, "REST and Web Services: In Theory and in Practice", Springer, 2011.

[6] B. Suda, " SOAP Web Services". University of Edinburgh, School of Informatics, Computer Science, 2003.

[7] H. Li , "RESTful Web Service Frameworks in Java", IEEE International Conference, Signal Processing, Communications and Computing (ICSPCC), 2011.

[8] A. Navarro, A.d.Silva, "A metamodel-based definition of a conversion mechanism between SOAP and RESTful web services", Computer Standards & Interfaces, pp.49-70,2016.

[9] F. Halili and M. Kasa, " Analysis and Comparison of Web Services Architectural Styles", In Proc. Book of International Conference of Information Technologies and their importance in the economic development, 701-712, Tirana. ISBN: 978-99956-59-13-4, 2011.

[10] K. Wagh and R. Thool, "A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host", Journal of Information Engineering and Applications, Vol 2, No.5, pp 12-16, 2012.

[11] M. Massé, "Designing Consistent RESTful Web Services Interface", Published by O'Reilly Media, ISBN: 978-1-449-31050-9, 2013.

[12] P. Giessler, et al, "Best Practices for the Design of RESTFul Web Services," Tenth International Conference on Software Engineering Advances, Barcelona, 2015.

[13] D. Qiu, et al, "Understanding the API usage in Java", Information and Software Technology, Elsevier, 2016, pp 81-100, 2016.

[14] J. Tihomirovs and J. Grabis, " Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics", Riga Technical University, 19, 92-97 P, 2016.

[15] F. Halili and E. Ramadani, "Web Services: A Comparison of Soap and Rest Services", Canadian Center of Science and Education, Modern Applied Science; Vol. 12, No. 3, 2018.

[16] A. Soni , et al ," API Features Individualizing of Web Services: REST and SOAP", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-9S, July, 2019.

[17] M. Govindaraju, et al,"Toward Characterizing the Performance of SOAP Toolkits", Fifth IEEE/ACM International Workshop on Grid Computing, Pittsburgh, PA, pp. 365-372, 2004.

[18] M. Gashti, "Investigating Soap And Xml Technologies In Web Service", International Journal on Soft Computing (IJSC) Vol.3, No.4, 2012.

[19] S.Malik, "A Comparison of RESTful vs. SOAP Web Services in Actuator Networks", 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, 2017, pp. 753-755, 2017.

[20] C. Kiama and L. Muchemi, "Comparative Study of REST and SOAP: Case of Registrar of Political Parties' Kenya", Trends in Distributed Computing, pp.105-116, 2014.

[21] A. Neumann and et al, "An Analysis of Public REST Web Service APIs", IEEE Transactions on Services Computing, 2018.

[22] R. Sinha, et al, "Design & Development of a REST based Web Service Platform for Applications Integration on Cloud", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 7, September 2014.

[23] R. Fielding and R. Taylor, "Principled design of the modern web architecture", ACM Trans. Inter. Tech., 2(2):115–150, 2002.

[24] V. Kumari, "Web Services Protocol: SOAP vs REST",International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 5pp.2467-2469, 2015.

[25] H. Hamad, et al, "Performance Evaluation of RESTful Web Services for Mobile Devices" Computer Engineering Department, Islamic University of Gaza, Palestine International Arab Journal of e-Technology, Vol. 1, No. 3, January 2010.

[26] M. Ali, et al, “Mobile Cloud Computing with SOAP and REST Web Services”, J. Phys.: Conf. Ser. 1018 012005, 2018.

[27] G. Serme, A. S. de Oliveira, J. Massiera and Y. Roudier, “Enabling Message Security for RESTful Services,” IEEE 19th International Conference on Web Services, Honolulu, HI, 2012, pp. 114–121, 2012. https://doi.org/10.1109/icws.2012.94

[28] Zinah.H.T, et al, “ An In-Depth Comparison Of Software Frameworks For Developing Desktop Applications Using Web Technologies”, Journal Of Southwest Jiaotong University, Vol. 54 No.4 Aug. 2019.