OPEN ACCESS
ISSN 2356-5462
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 10, No. 2, Dec 2024. pp. 214-224
doi: 10.21108/ijoict.v10i2.1030

JSON-Based RSS as an Alternative to XML-Based RSS on AJAX-Based Aggregator Websites

Dody Qori Utama[1*], Yanuar Firdaus[2], Dana Sulistiyo Kusumo[3]

[1,3]*School of Computing, Telkom University*
Indonesia

[2]*International Islamic University of Indonesia*
Indonesia

[*] dodyqori@gmail.com

Abstract

With the advancement of technology, websites have become widely used mediums for publishing content. Really Simple Syndication (RSS), also known as Rich Site Summary, is a technology that enables people to share website content more quickly and easily. eXtensible Markup Language (XML) is a versatile, free markup language used to describe various types of data. JavaScript Object Notation (JSON) is a format for exchanging data between computers; it features a simple structure that is easily readable by humans. Asynchronous JavaScript and XML (AJAX) is a technique that enhances website interactivity by allowing web applications to communicate with the server asynchronously. In this study, the author compares a proposed JSON-based method with existing technologies based on three criteria: speed of information creation, data size, and data reading speed. The proposed method demonstrated better performance than the existing technologies.

Keywords: RSS JSON, AJAX, RSS XML

I. INTRODUCTION

Along with the development of technology, websites have become one of the media that are widely used as a medium to publish something. Website content is also increasingly diverse and complex [1]. Currently, many people think about sharing website content and also publishing their websites. RSS is an answer to the needs of many people.

Really Simple Syndication or Rich Site Summary (RSS) is a technology that allows people to share content from a website more quickly and easily [2]. RSS allows everyone to share content from any website that provides RSS services without having to bother thinking about the differences in technology on existing websites. RSS is website content that is shared so that it can be read easily by RSS Reader. RSS is created in XML format so that RSS can be read and used even though there are differences in the technology used.

XML is a data format used in RSS. XML, which stands for eXtensible Markup Language, is a data format in the form of markup that is often used to exchange data within a system or between systems [3]. XML is an easy way to exchange data even though the systems that are exchanging have different technologies. XML is the format recommended by the W3C (World Wide Web Consortium) for exchanging data [4]. Currently, XML is

Received on 19 Dec 2024. Revised on 20 Dec 2024. Accepted and Published on 10 Jan 2024.

the only format used to share website content via RSS. However, over time a new format has emerged called JSON which has the same functionality as XML but with its own advantages over XML.

JSON (JavaScript Object Notation) is a subset of the JavaScript programming language that is also used for data exchange [5]. JSON began to be introduced as an alternative format besides XML in transmitting data after AJAX technology developed. Although JSON is a subset of JavaScript, JSON is a data format that does not depend on a particular language. The JSON format is specified in RFC 4627. JSON has advantages in the simplicity of writing syntax and speed of data parsing in some cases compared to XML and its smaller data size. Currently, JSON is widely used as an alternative format besides XML. RSS, which is a technology for sharing content, also allows the use of JSON as a data exchange format. However, until now there has been no website content sharing technology that uses JSON as a medium of exchange. By utilizing JSON technology which has these advantages, the author proposes the use of JSON in a content sharing format.

II. BASIC THEORY

A. *RSS*

RSS, which stands for Really Simple Syndication or Rich Site Summary, is a technology that was born due to the rapid development of website technology. RSS is a technology that was developed to create ease in sharing information or content [6]. With RSS, information and content can be exchanged even though the system architecture is different from each other. RSS is a development of RDF technology that has been developed since 1995. RSS has currently reached version RSS 2.0 [7].

RSS is widely used on websites. One of the biggest users of RSS is news service providers. News providers generally use RSS to provide news to their service users. With RSS, users of news website services can subscribe to news like subscribing to newspapers. In addition to news websites, one of the users of RSS technology comes from blog users. With RSS, fellow blog users can exchange information between one blog and another or even with another website. Sharing information is now easier with RSS. RSS is built using the eXtensible Markup Language (XML) markup language.

RSS 2.0 is a development of the previous version of RSS. RSS currently has 4 versions, namely RSS 0.9, RSS 1.0, RSS 1.1 and RSS 2.0. RSS 2.0 has the advantage of ease of use compared to previous versions of RSS. RSS 2.0 has ease in writing syntax compared to previous RSS. RSS 2.0 has now become a world standard as a technology that functions to share content and information [8], [9].

RSS 2.0 has many advantages compared to other methods. The advantages of using RSS 2.0 are:

- It saves bandwidth because the files are delivered in small XML format [10].
- Easier to use, simple and time-saving. RSS makes itself automatically updateable. With RSS we don't need to visit our favorite websites to find out the latest changes. Simply by subscribing to RSS from our favorite website, we will automatically receive news if there is an update of content or information [8].

B. *XML*

eXtensible Markup Language (XML) is a versatile free markup language used to describe various types of data. XML is a markup language where users are free to define their markup elements. XML is a versatile markup language because XML has many functions. XML is usually used for:

- Forming certain data structures to facilitate data exchange.
- Save data.
- As a data exchange format on many systems.
- Defines the contents of a document.

XML is a powerful markup language because it can be used anywhere without recognizing differences in system architecture. Because it does not recognize differences in architecture, XML is usually used to exchange information for both software and hardware applications [11].

C. JSON

Javascript Object Notation (JSON) is a format for exchanging data between computers. JSON has a simple format and can be easily read by humans. JSON, which acts as a data exchange medium, is an alternative to the currently frequently used format, XML. JSON was originally a subset of the JavaScript language for modeling objects. However, along with the development of technology, JSON has now become an independent and robust data exchange format. JSON can now be used in many programming languages [12], [13].

D. AJAX

Asynchronous Javascript and XML (AJAX) is a technique to make a website more interactive. AJAX allows a web application to interact with the server asynchronously. With AJAX, a website application will be more responsive and increase the user experience of its users. AJAX is not a programming language. AJAX is a combination of existing techniques to create something new. The techniques and technologies involved in AJAX are:

- *Javascript*
- *XMLHttpRequest Object*
- *Server Side Language*
- XML / JSON

AJAX is basically a technology to make asynchronous connections with the server. With asynchronous connections, it is possible for someone to do many things at one time unlike in classical techniques. Asynchronous connections in AJAX are the responsibility of the XMLHttpRequest object [8], [12], [14], [15], [16].

III. SOFTWARE DESIGN

A. Software Illustration

The software in my final project is divided into two important parts, namely RSS generator which functions to produce RSS and RSS Reader which functions to read existing RSS. RSS Generator is installed on the server while RSS Reader is installed on the client.

B. RSS Generator Architecture

RSS Generator that functions to produce RSS according to the specified format. RSS generator works by reading the existing news database and processing it into an RSS based on XML and JSON. The description of the RSS Generator architecture is shown in Fig. 1.

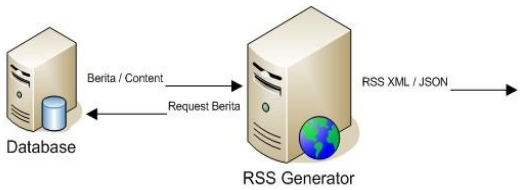

Fig. 1. RSS Generator

C. Software Illustration

RSS Reader functions to read RSS generated by an RSS Generator. RSS Reader reads RSS that is still in XML or JSON format. After reading RSS that is still in XML / JSON format, RSS Reader also serves to translate

it into a form that is understood by users who use RSS Reader. Fig. 2 is an architectural description of RSS Reader.

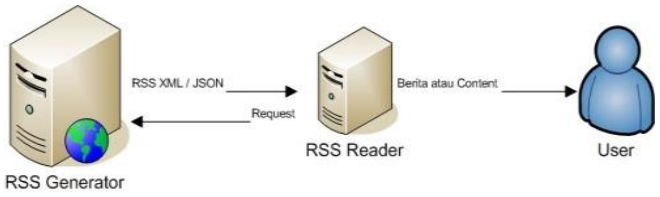

Fig. 2. RSS Topology

D. RSS XML Data Format

The default RSS is an XML file. RSS XML consists of various markup tags, including title, which functions to display the news title, description, which contains a description of the content and news, and other standard RSS markup tags. Fig. 3 is an example of standard RSS XML:

RSS Example

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
 <channel>
  <title>News Center</title>
  <link>http://localhost</link>
  <description>This is a News Center Website</description>
  <item>
   <title>Chris John's Rematch in Indonesia Rejected</title>
   <link>http://localhost:8080/news_center/de tailberita.php?id_ber=103 </link>
   <description><p><strong>Pliersgroan</strong> -     Just play draw with Rocky Juarez in
Houston, Chris John asked for a <em>rematch </em>to be held in Indonesia.
   However, this request later had to be rejected.<br ... <br /> </description>
   <enclosure url="http://localhost:8080/news_center/news_i mage/103 " length="10240"
type="image/jpeg"/>
  </item>
  </channel>
</rss>
```

Fig. 3. RSS Example

JSON RSS Example

```
{"title":"News Center",
"link":"http://localhost",
"ddescription":"IThis is a News Center Website",
"channel":{
      "item":[{
            "title":" Chris John's Rematch in Indonesia Rejected ",
            "link":"http://localhost:8080/news_center/detailberita.php?id_ber=103",
            "description":"<p><strong>Tangerang</strong> - After drawing with Rocky Juarez
in Houston, Chris John calls for a rematch </em>in Indonesia. However, this request was
later rejected.<br ... ",
            "enclosure url":"http://localhost:8080/news_center/news_i mage/103"}, ]}
}
```

Fig.4. JSON RSS Example

E. JSON RSS Data Format

JSON-based RSS is basically non-existent. In this final assignment, JSON-based RSS is a form of transformation and refinement of the existing RSS format. The difference between RSS JSON and RSS XML data formats is more about writing. Writing RSS JSON is simpler than RSS based on XML. In addition, RSS XML does not allow writing HTML syntax so that conversion is needed in RSS XML. In RSS JSON, the HTML tag conversion process is not required. The conversion process in RSS JSON is only needed to convert quotation marks (` ” `) to Backslash quotation marks (` \” `). Fig. 4 is an example of an RSS JSON created by the author.

IV. TEST RESULTS AND ANALYSIS

A. Analysis Large Data Testing of RSS XML and RSS JSON

Fig. 5 is the test data for large JSON and XML data in RSS format. Based on the test data, it can be seen that RSS JSON always has a smaller size than RSS XML. This is due to the use of less syntax to produce an RSS by JSON compared to XML. Based on the existing RSS JSON design, RSS JSON has rules that save writing tags, and writing a lot of data (Array) used in news item elements.

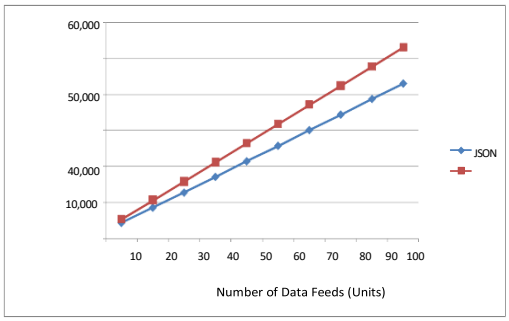

Fig. 5. Comparison Table of RSS XML and RSS JSON in Large Data Sizes

The average RSS JSON requires 400-450 bytes to generate a news feed. While RSS XML requires an average of 500-550 bytes to produce a feed. The difference of 100 bytes or 100 characters, in addition to being caused by the writing syntax, is also caused by the influence of the conversion results of the markup language syntaxes in XML, while in JSON the conversion process does not need to be done.

The difference in size between RSS JSON and XML is not significant, which is only about 100 bytes for one feed. However, for a large-scale news website such as Google News and Detik which has millions of feed subscribers per day, it will save bandwidth costs and reduce traffic to the news website. RSS JSON has a size of 0.8 times that of RSS XML. Of course, there will be a bandwidth saving of 20%. In addition, RSS users also come from people who rely on mobile technology to read the news. The smaller the size, the more resources will be saved for those who always want to be updated with the latest news.

B. Generate Time Testing Analysis RSS XML and JSON

The following is the test result data from RSS XML and JSON where the results are separated into two different tables so that they can be compared. From the results of the experiment above, it can be seen that for small data ranges (between 10 - 30 data), XML and JSON have almost the same performance where JSON is very slightly superior to XML. This is because the JSON file size is smaller than the XML file size (4:5) so that it speeds up the writing of JSON data compared to XML. However, in general, for small news feed sizes, RSS JSON and RSS XML have almost the same performance.

When the data feed enters the amount above 40 data, the change in JSON and XML generation time both experience a spike compared to when the data is still below 30 data. This spike is more due to the server having

to start adding new data cache on the server because it has to accommodate larger data so that additional time is more intended for adding cache on the server. JSON data begins to show a faster time than XML. This is more due to the process of generating RSS XML, there is a process to change or convert markup language tags that are generally used in HTML so as not to interfere with the existing XML format.

While in JSON the conversion process is not needed because the markup language tags will not mess up the JSON data format. The conversion process and less data in RSS JSON make JSON data superior to RSS XML. As the data grows larger, XML RSS data shows a more inconsistent generation time compared to JSON RSS. This is more due to the HTML tag conversion process where the amount of converted data is uncertain. Unlike JSON, which does not require a conversion process.

In the comparative experiment of RSS XML and JSON generation time, it is also sometimes found that the process of generating more data feeds takes less time than data with a smaller number of feeds. This is more due to the server delay time which is sometimes inconsistent due to the large number of requests to the server and the data flowing through a network. This can be seen in the JSON and XML generation results table, where there is a generation time that is much larger than the data.

Overall, RSS JSON is superior to RSS XML in terms of generating data. RSS JSON speed is superior between 1-1.5 times the speed of RSS XML. In this test, RSS JSON is superior in milliseconds. Overall, it is not significant because it is done within the scope of an intranet network and the request to the server is not too large. The XML and JSON generation time which is in milliseconds will certainly increase along with the size and complexity of the network and web and the increasing number of service requests to the server tasked with generating feeds.

C. Analysis Testing Generate RSS XML and RSS JSON with Many Users Accessing Simultaneously

The following is the test result data from RSS XML and JSON where the results are separated into two different tables so that they can be compared. First test results in Table I show the RSS JSON accessed by many

TABLE I
RSS JSON ACCESSED BY MANY USERS

Total JSON Time	Number of Accessors
0.005985022	1
0.01369119	2
0.0198014	3
0.0255566	4
0.03256965	5
0.0379646	6
0.044247	7
0.04933	8
0.056554	9
0.062397	10

TABLE II
RSS XML ACCESSED BY MANY USERS

Total XML Time	Amount Accessor
0.00784111	1
0.01353621	2
0.0187922	3
0.0295722	4
0.03587341	5
0.0415485	6
0.050316	7
0.05749	8
0.062919	9
0.067983	10

users. Testing is carried out on 1-10 users at the same time and the total time obtained by each user is taken. The amount of data is 10 feeds. Table II shows the RSS XML performance accessed by many users.

Testing is carried out on 1-10 users at the same time and the total time obtained by each user is taken as shown by Fig. 6. The amount of data is 10 feeds. Test results in Table III show RSS JSON accessed by many users.

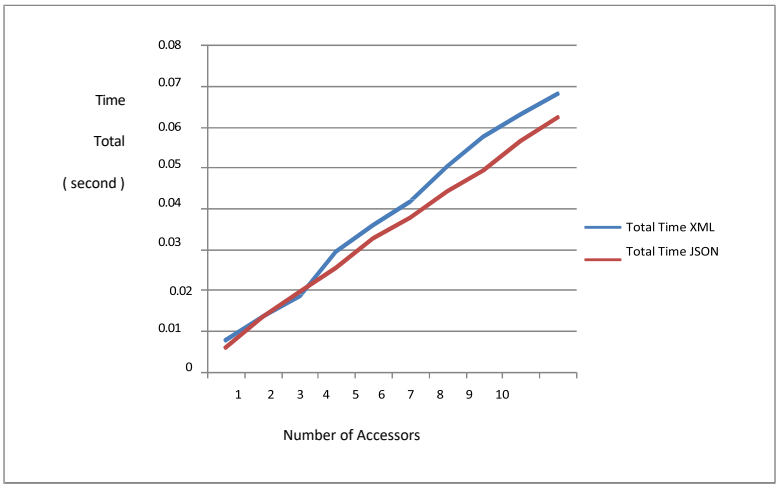

Fig. 6. Comparison XML and JSON Total Time

TABLE III
RSS JSON ACCESSED BY MANY USERS (2ND TEST)

Total JSON Time	Amount Accessor
0.016618013	1
0.0398941	2
0.0629921	3
0.0860832	4
0.11095524	5
0.1374192	6
0.163908	7
0.191585	8
0.222737	9
0.247041	10

TABLE IV
RSS XML ACCESSED BY MANY USERS (2ND TEST)

Total XML Time	Amount Accessor
0.023257971	1
0.04344106	2
0.0691612	3
0.0903323	4
0.12414122	5
0.1527653	6
0.176076	7
0.19851	8
0.221113	9
0.246033	10

Testing was conducted on 1-10 users at the same time and the total time obtained by each user was taken. The amount of data was 50 feeds. Test results in Table IV show the RSS XML accessed by many users. Testing

was conducted on 1-10 users at the same time and the total time obtained by each user was taken as shown in Fig. 7. The amount of data was 50 feeds.

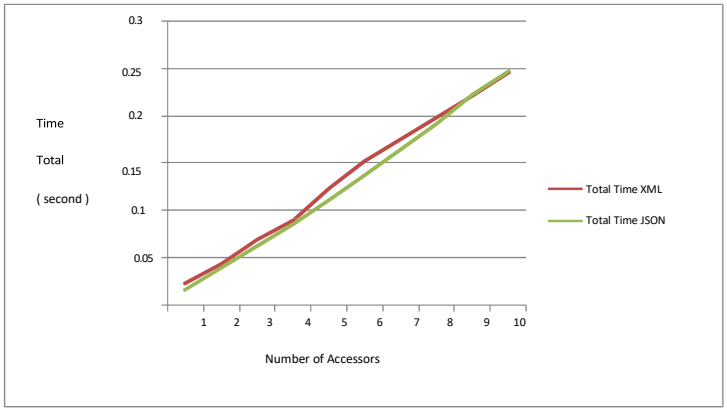

Fig. 7. Comparison XML and JSON Total Time under Second Test

Third Test Results are shown in Table V for RSS JSON Accessed by Many Users. Third Test Results are shown in Table VI for RSS XML Accessed by Many Users. Fig. 8 summarizes the comparative results.

TABLE V
RSS JSON ACCESSED BY MANY USERS (3RD TEST)

Total JSON Time	Amount Accessor
0.044275045	1
0.08985019	2
0.1271312	3
0.1589332	4
0.20828032	5
0.2544284	6
0.290699	7
0.341233	8
0.393147	9
0.440752	10

TABLE VI
RSS XML ACCESSED BY MANY USERS (3RD TEST)

Total XML Time	Amount Accessor
0.119044065	1
0.1837182	2
0.2250342	3
0.2932782	4
0.32530618	5
0.3719523	6
0.439178	7
0.483563	8
0.119044065	1
0.1837182	2

Based on the 3 experimental data above, the performance of RSS JSON and XML is almost the same in the small to medium data range with JSON almost always superior. RSS XML can keep up with RSS JSON more because the markup tag conversion process is done less because when RSS data is accessed simultaneously, the server will temporarily store the cache so that the conversion process only needs to be done by the user who first opens it, while the next user who.

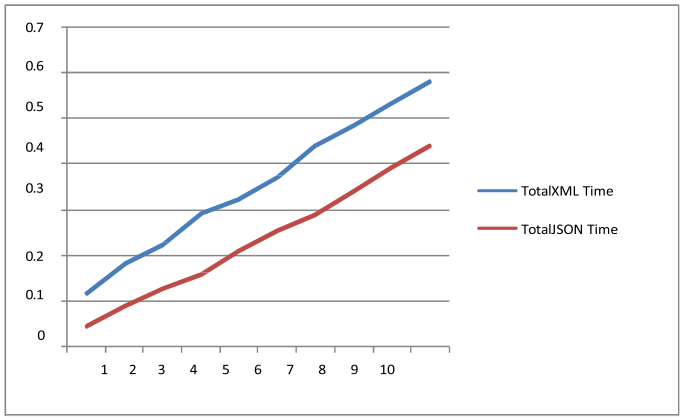

Fig. 8. Comparison XML and JSON Total Time under Third Test

Opening at almost the same time can still utilize the cache that has just been formed and has not been deleted. When the data starts to grow bigger (100 data), RSS XML starts to increase leaving RSS JSON. This is more because the bigger the data, the more cache is needed, so RSS XML will require more server resources than RSS XML.

In testing where RSS is accessed simultaneously by users, it also sometimes has a better generate time than the usual generate test (test 2), This usually happens in RSS XML. This is more due to the advantage of cache which causes the server not to have to do the same process over and over again. In RSS JSON the average RSS generate time for many users is almost the same as the RSS generate time. This is because the speed of generating RSS JSON is almost the same as the time to use cache resources.

The increase in the number of people accessing RSS also affects the server's working time. The more people accessing an RSS, either JSON or XML-based, the more the user's RSS receiving time increases. It can be concluded that the server's working time in handling RSS requests at the same time is directly proportional to the number of people making requests. In large amounts of data, JSON is always superior in handling RSS requests. This will be beneficial for media that use RSS a lot as a medium for sharing content if accessed by many users simultaneously. Based on the above tests, RSS JSON has a speed between 1-1.25 times faster than RSS XML.

D. Analysis RSS XML and RSS JSON Parsing Time

Fig. 9 is the parsing data for each RSS in the AJAX-based reader embedded in the client. Based on the test data above, the parsing results between RSS JSON and RSS XML produce results that the parsing time between XML and JSON is almost the same. Even if there is a difference, the difference is only in the nano seconds

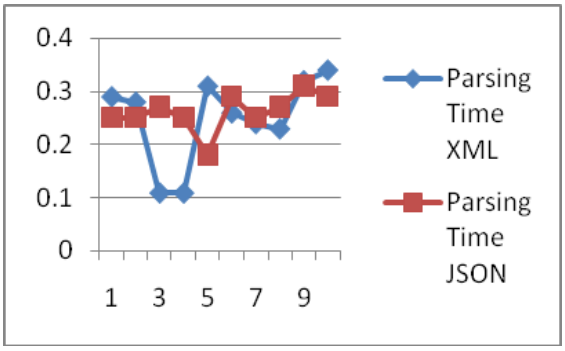

Fig. 9. Parsing Time Comparison Graph RSS XML and RSS JSON

range. Such a small difference is insignificant because parsing is only done on the client side. Even in the experimental data, sometimes larger data is parsed faster than smaller ones. This is more because the difference is very small (in nano seconds) so that the computer has difficulty calculating the difference.

V. CONCLUSION

Based on the experiments and analysis that have been discussed and carried out, the following points can be concluded as follows. JSON RSS has better performance because JSON RSS is superior in terms of size and generation time compared to conventional RSS. RSS JSON is more stable in data size when carrying large amounts of data. RSS JSON is a worthy alternative for a large-scale content sharing system with many users. RSS JSON is not recommended for use on small-scale content sharing systems (blogs, personal websites, small-scale institutional websites). RSS JSON is worthy of being the main alternative to replacing a content sharing system with target users who want to save resources.

REFERENCES

[1] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding website complexity," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, New York, NY, USA: ACM, Nov. 2011, pp. 313–328. doi: 10.1145/2068816.2068846.

[2] M. J. W. Lee, C. Miller, and L. Newnham, "RSS and content syndication in higher education: subscribing to a new model of teaching and learning," *EMI Educ Media Int*, vol. 45, no. 4, pp. 311–322, Dec. 2008, doi: 10.1080/09523980802573255.

[3] "Using XML," in *Practical Mono*, Apress, pp. 241–272. doi: 10.1007/978-1-4302-0097-0_9.

[4] L. Mignet, D. Barbosa, and P. Veltri, "The XML web," in *Proceedings of the twelfth international conference on World Wide Web - WWW '03*, New York, New York, USA: ACM Press, 2003, p. 500. doi: 10.1145/775152.775223.

[5] K. Afsari, C. M. Eastman, and D. Castro-Lacouture, "JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange," *Autom Constr*, vol. 77, pp. 24–51, May 2017, doi: 10.1016/j.autcon.2017.01.011.

[6] R. J. Glotzbach, J. L. Mohler, and J. E. Radwan, "RSS as a course information delivery method," in *ACM SIGGRAPH 2007 educators program*, New York, NY, USA: ACM, Aug. 2007, p. 16. doi: 10.1145/1282040.1282057.

[7] L. Attouche, M.-A. Baazizi, D. Colazzo, G. Ghelli, C. Sartiani, and S. Scherzinger, "Validation of Modern JSON Schema: Formalization and Complexity," *Proceedings of the ACM on Programming Languages*, vol. 8, no. POPL, pp. 1451–1481, Jan. 2024, doi: 10.1145/3632891.

[8] Danny Ayers and Andrew Watt, *Beginning RSS and Atom Programming*. John Wiley & Sons, 2005.

[9] D. Johnson, *RSS and Atom in Action: Web 2.0 Building Blocks*. USA: Manning Publications Co., 2006.

[10] G. Singh and S. Sahu, "Review on "Really Simple Syndication (RSS) Technology Tools"," in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, IEEE, Feb. 2015, pp. 757–761. doi: 10.1109/CICT.2015.109.

[11] J. Tekli, "An Overview on XML Semantic Disambiguation from Unstructured Text to Semi-Structured Data: Background, Applications, and Ongoing Challenges," *IEEE Trans Knowl Data Eng*, vol. 28, no. 6, pp. 1383–1407, Jun. 2016, doi: 10.1109/TKDE.2016.2525768.

[12] R. Lal, *Professional Web Widgets with CSS, DOM, JSON and Ajax*. GBR: Wrox Press Ltd., 2009.

[13] Jeremy Keith, *DOM Scripting*. Apress, 2005. doi: 10.1007/978-1-4302-0062-8.

[14] S. Holzner, *Ajax bible*. John Wiley & Sons, 2007.

[15] S. Holzner, *Ajax for dummies*®. USA: John Wiley & Sons, Inc., 2006.

[16] Lee Babin, *Beginning Ajax with PHP*. Berkeley, CA: Apress, 2007. doi: 10.1007/978-1-4302-0253-0.