XML versus JSON

Introduction

This research paper describe two raw text formats: called Extensible Markup Language (XML) and JavaScript Object Notation (JSON). In the late 1990s humanity reached a new milestone in computing: the “one person many computers” era. This era saw improvement in backend technologies. The main players were web services technologies, who rapidly developed their techniques for data sharing between different software architectures. Most service providers adopted one or both documents data structures based on their underlying system architecture. Both XML and JSON are designed for facilitating constant and reliable data transmission between service layers and platforms. However, each has its own benefits and drawbacks. Section 2 and 3 of this paper will discuss XML, and section 4 and 5 will discuss JSON. Section 6 will then compare them and describe their optimal usage.

About XML

In the second era of computing, World Wide Web developers, recognized the first problem with a large number of web sources: the universal data exchange between heterogeneous systems. Charles Goldfarb, Ed Mosher and Ray Lorie invented a solution: the General Markup Language (GML), the first structural document with tags. Later it became known as Standard General Markup Language (SGML), and was ISO adopted in 1986. The biggest IT companies such as Oracle, Microsoft and IBM voted to implement a cross-platform database solution using XML data layout. So XML became the independent document structure, and provided excellent communication between dissimilar platforms and programming languages. Goldfrab defined XML as, “the holy grail of computing, solving the problem of universal data interchange between dissimilar system” (Papaleo, 2009). The current XML version is 2.0. However, the previous 1.0 version is still widely used by different platforms.

Structure of XML

XML document have a tree structure. Each document must have only one root element. Each element has a starting and closing tag with angle brackets, and the developer can define the name of each element. Furthermore, each element can have nested elements and multiple attributes. Attributes describe data within an element tag (Lindo, S.)

e.g <myfirstelement value=’myfirstvalue ’> my first content </myfirstelement>

Finally, the well-structured XML document contains the following characteristics such as

* Declaration: <?xml version="1.0" encoding="utf-8" ?>
* Comment : <!-- comment -->
* Instruction: <?xml-stylesheet href="style.css" type="text/css" ?>
* Namespaces: <doc xmlns="http://sabi11.ddns.net/2016/namespace"  
  xmlns:prefix="http://sabi11.ddns.net/2016/other">

About JSON

Douglas Crockford developed JSON around 2011. JSON stands for JavaScript Object Notation. It is a machine-readable data-representation format with hierarchical structure (json-csv.com, 2016). It relies on value pairs within parent child relationships. It is derived from JavaScript programming languages. The REST web service mainly uses JSON. JSON has been growing along with AJAX communication mechanisms because it can be processed faster and transferred quicker over the internet (jsonapi.org, 2016).

Structure of JSON

The JSON format is based on key-value pairs encapsulated with double or singe quotes and separated by a colon. It can have a nested elements called an array, which has square brackets. The following is an example taken from (json-schema.org, 2016)

The JSON format as was mentioned it is based on key-value pairs encapsulated with double or single quote and separated by colon. It can have a nested element, it is called array which represents a square brackets.

{

"$schema": "http://json-schema.org/draft-04/schema#",

"title": "Product",

"description": "A product from Acme's catalog",

"type": "object",

"properties": {

"id": {

"description": "The unique identifier for a product",

"type": "integer"

}

},

"required": ["id"]

}

[ref: http://json-schema.org/example1.html]

Comparison

Simple Object Annotation Protocol (SOAP) originally used XML, and RESful Web Service widely uses JSON. Recently the XML data structure was dropped by newer Web Services such as Twitter, Facebook, Instagram and Foursquare. These companies’ front-end interfaces are now built on a JavaScript driven framework with asynchronous communication such as AJAX. Processing JSON format is faster and easier then XML. JSON, along with an Application Programming Interface (API), is much more lightweight for moving large amounts of data between database and users. This is especially important when an application is rendered on a portable device. Developers must consider many aspects such as screen size, resolution, network bandwidth, computation power and battery power consumption.

So it seems JSON has won against XML in the mobile world. However, there are great numbers of other available systems that still use XML rather than JSON. This because XML has several advantages over JSON. For example, XML contains much more information such as namespace and attribute. In other words, when the service focuses on a document rather than raw data, XML has namespace schemas and entities which help to identify the origin of the XML documents. In that sense, XML is more reliable than JSON because JSON has no schemas or attributes. Furthermore, XML has another purpose within systems: it can represent underlying configuration files and meta-data in almost every type of service, such as J2EE configuration files, Android configuration and layout files, etc.

Nowadays, almost every device is connected to a network. Subsequently, the internet ha a fundamental layout of large distributed network. Therefore, both document structures can be easily transferred via Hyper Text Transfer Protocol (HTTP). The main reasons for the popularity of HTTP are: All programming languages contain a library to parse or generate either XML or JSON document format based on the requirements; Firewalls do not block HTTP, and HTTP can be combined with SSL to provide a secure peer-to peer connection (HTTPS).

Conclusion

This pare has discussed XML and JSON. The first section gave on overview of the evolution of web services. The overview also described the problem of data exchange between dissimilar platforms. XML and JSON solve this problem by bridging communication problems. XML was the first standard markup language and completely platform-independent solution. It allows vendors to send and receive any type of data to other vendors. The next section outlined the major characteristics of well-structured XML format with tree-structured layout and elements and namespaces. The next section explained the JSON data type. The last section compared XML with JSON and discussed their usage, data handling, internet relationship as a communication medium and a data transfer protocol.

References

[1] S. Kinsley, \Ubiquitous computing: Mark weisers vision and

legacy," 12/03/2012. [Online]. Available: http://www.samkinsley.com/2010/03/12/

ubiquitous-computing-mark-weisers-vision-and-legacy/

[2] S. Lindo, \Xml vs. json - a primer," 11/07/2013. [Online]. Available: http://www.

programmableweb.com/news/xml-vs.-json-primer/how-to/2013/11/07

[3] L. Papaleo, \Introduction to xml and its applications," p. 03/02/2016, 2016. [Online]. Available:

https://www.academia.edu/1041294/Introduction to XML and its Applications

[4] D. Crockford, JavaScript: The Good Parts, 2008.

[5] S. Klabnik, Y. Katz, D. Gebhradt, T. Kellen, and E. Resnick, \A speci\_cation for building apis

in json," 11/12/2015. [Online]. Available: http://jsonapi.org/

[6] G. Google, \Json schema." [Online]. Available: http://json-schema.org/

[7] \Example data," 2016. [Online]. Available: http://json-schema.org/example1.html

[8] M. Sporny, \Web services: Json vs. xml," 22/11/2010. [Online]. Available: http:

//digitalbazaar.com/2010/11/22/json-vs-xml/

[9] S. Francia, \Rest vs. soap, the di\_erence between soap and rest," 15/01/2010. [Online].

Available: http://spf13.com/post/soap-vs-rest