

# **Lab 2**

## **XML vs JSON**

**Nicholas Randles – B00058026**

## **XML vs JSON**

XML and JSON are both formats used to transfer data over the web. XML has been around longer than JSON. In 1996, XML was created by W3C and in 2002 JSON was identified by Douglas Crockford[1]. XML and JSON are both less data-heavy than HTML. This is because HTML contains data such as images which are irrelevant to us when we are just looking for pure data. We are able to combat this problem by using web services. A web service is a specific function call to a server. Ideally, a web service that suits the applications specific needs is best. An advantage of web services is that they focus on a specific part of data. This means the application does not need to waste data usage and processing power on large amounts of data which is mostly irrelevant. Instead, the application receives a small amount of data that is focused on its needs. JSON and XML are both return formats which are readable by humans and machines[2]. The main purpose of XML is document markup. This is achieved by applying tags to the text which is laid out in a tree-like structure. The main purpose of JSON is structure data transferring. It achieves this by directly describing objects, arrays, strings, booleans and numbers. Neither format is clearly better than the other as they both are more suited to carry out their own purposes.

XML has a number of advantages over JSON. Some of them include:

- With XML you can simply put metadata inside the tags as an attributes. In order to achieve this in JSON you would have to make the entity an object and add the attributes to the object. The XML approach is considered better as the JSON way can cause confusion as the newly created object is not from the client program.
- XML has a stronger naming convention which allows it to avoid name conflicts. The programmer has the ability to name two entities the same thing by using prefixes. This is useful for situations for when two entities should have the same name but relate to something different.
- XML is normally show in an organised tree-like structure in most browsers which allows the user to easily see the elements clearly. They can collapse each tree element when they wish. This formatting feature helps accommodate tasks such as debugging.
- Another advantage XML has over JSON is its ability to easily represent mixed content. This can be simply done by wrapping the child tags around the marked-up text. There is no simple way for JSON to indicate markup and you may have to abuse its format to do so.

JSON has a number of advantages over XML. Some of them include:

- JSON is a lot less verbose than XML. Unlike XML which uses opening and closing tags, JSON uses name/value pairs. A JSON file which contains the same information as an XML file is a lot smaller which makes it faster to process and transmit. They use less resources which make them a better option for mobile devices as they tend to be slower than desktops.

- Another advantage JSON has over XML is its ability to describe its objects and arrays and to directly map them to corresponding data structures. This is a lot harder to achieve in XML and requires additional code.
- Another big advantage JSON has over XML is the fact that it is a part of JavaScript. This means that it is very easy to work with JSON using JavaScript. JSON also provides many tools to help integrate JSON processing with other programming languages.

## XML vs JSON

<pre> &lt;empinfo&gt;   &lt;employees&gt;     &lt;employee&gt;       &lt;name&gt;Scott Philip&lt;/name&gt;       &lt;salary&gt;£44k&lt;/salary&gt;       &lt;age&gt;27&lt;/age&gt;     &lt;/employee&gt;     &lt;employee&gt;       &lt;name&gt;Tim Henn&lt;/name&gt;       &lt;salary&gt;£40k&lt;/salary&gt;       &lt;age&gt;27&lt;/age&gt;     &lt;/employee&gt;     &lt;employee&gt;       &lt;name&gt;Long yong&lt;/name&gt;       &lt;salary&gt;£40k&lt;/salary&gt;       &lt;age&gt;28&lt;/age&gt;     &lt;/employee&gt;   &lt;/employees&gt; &lt;/empinfo&gt; </pre>	<pre> { "empinfo" :   {     "employees" : [       {         "name" : "Scott Philip",         "salary" : £44k,         "age" : 27,       },       {         "name" : "Tim Henn",         "salary" : £40k,         "age" : 27,       },       {         "name" : "Long Yong",         "salary" : £40k,         "age" : 28,       }     ]   } } </pre>
--	---

Figure 1: XML vs JSON code example [4]

The “Internet of things” is one of the most expected changes in the internet’s future. This refers to the addition of non-computer devices to the internet such as televisions, watches and refrigerators. This trend is expected to continue to grow dramatically over the next few year with the number of devices expected to reach a range between 26 billion and 200 billion[1]. Most of these devices are smaller and less powerful than regular computers which means they need a more lightweight approach to transfer and process data. This makes JSON a viable option as it is lightweight and faster than XML. These new devices may benefit more by being able to read JSON due to its increasing growth and popularity.

The recent large increase in popularity of server-side JavaScript frameworks such as Node.js and NoSQL databases such as MongoDB has made full-stack JavaScript development a serious player. This will help JSON’s future as these applications use JSON. JSON normally helps to make them lightweight and fast which makes these types of applications attractive to developers. Another trend in application development is the move away from SOAP and

towards REST[3]. Unlike SOAP which only uses XML, Rest uses JSON and XML which will also help increase JSON usage.

After analysing these trends I believe it is safe to say that JSON will continue its large growth over the web as XML continues to decline. XML will continue to exist but it will not be as important as it once was. It will still have to be used for certain applications such as ones that use SOAP. Although application development has been increasingly changing over from SOAP to REST development which means applications dependent on XML are decreasing. Full-stack JavaScript is a growing trend and is showing no signs of decreasing mainly down to its high performance. Since full-stack JavaScript applications use JSON this will also contribute to the growth of JSON. I believe that all of these trends will make JSON much more widely used over the web than XML.

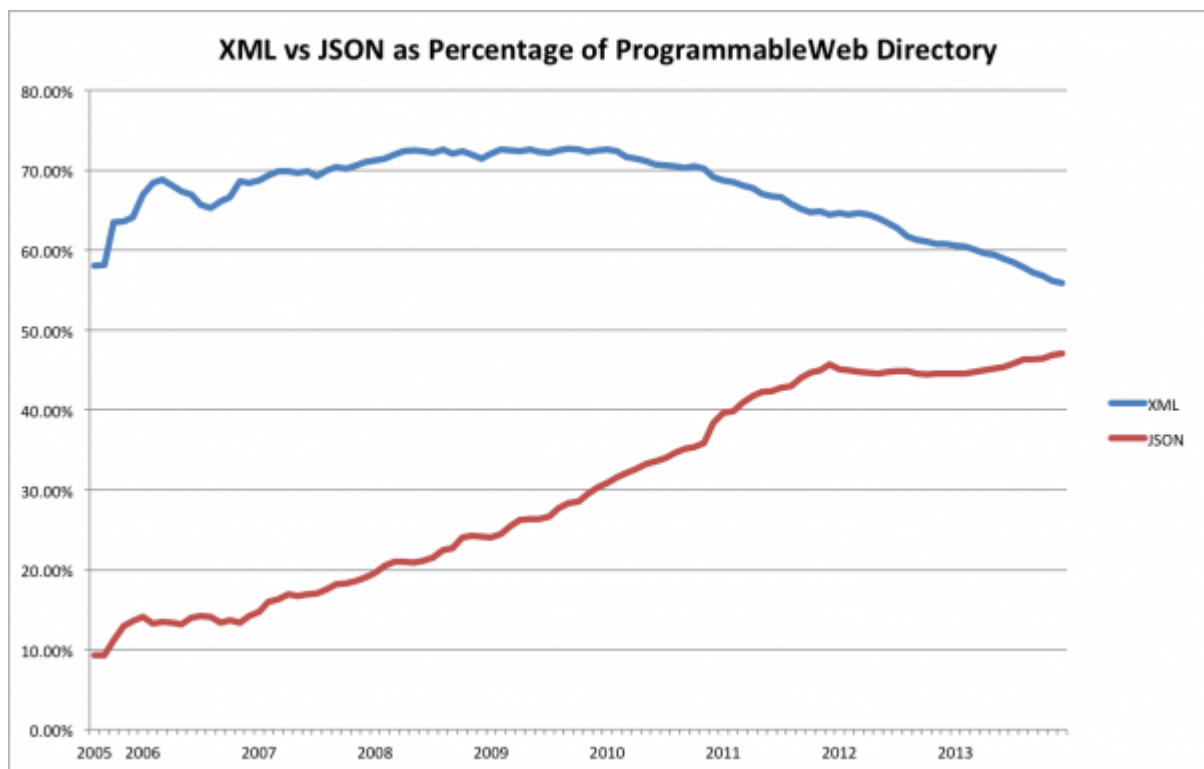


Figure 2: Popularity of XML vs JSON [5]

## **References**

- [1] "XML Vs JSON " [online].  
Available: [http://www.cs.tufts.edu/comp/150IDS/final\\_papers/tstras01.1/FinalReport/FinalReport.html#r12](http://www.cs.tufts.edu/comp/150IDS/final_papers/tstras01.1/FinalReport/FinalReport.html#r12). [Accessed 2/2/2016].
- [2] "XML Vs JSON Based Web Services: Which is the Best Choice? | Segue Technologies - Custom Software Development, Professional Website Design, Information Technology " [online]. Available: <http://www.seguetech.com/blog/2013/04/01/xml-vs-json-web-services-best-choice>. [Accessed 2/2/2016].
- [3] "XML Vs. JSON - A Primer | ProgrammableWeb " [online].  
Available: <http://www.programmableweb.com/news/xml-vs.-json-primer/how-to/2013/11/07>. [Accessed 2/2/2016].
- [4] "Web Architecture - Mechanism and Threats " [online].  
Available: <http://www.slideshare.net/materaj/web-architecture-mechanism-and-threats>. [Accessed 2/2/2016].
- [5] "JSON's Eight Year Convergence with XML | ProgrammableWeb " [online].  
Available: <http://www.programmableweb.com/news/jsons-eight-year-convergence-xml/2013/12/26>. [Accessed 2/2/2016].