JSON and PESC XML

4/24/2014

# Conversion from XML to JSON:

<http://www.xml.com/pub/a/2006/05/31/converting-between-xml-and-json.html>

Complete Syntax:

<http://json.org/>

Conversion rules:

<http://badgerfish.ning.com/>

We just need this for JSR 353 and I think we are done on this one.

More links of highly technical nature:

<http://examples.javacodegeeks.com/enterprise-java/rest/resteasy/json-example-with-resteasy-jaxb-jettison/>

<http://www.javaworld.com/article/2074650/core-java/javaone-2012--jsr-353--java-api-for-json-processing.html>

JavaDoc details - JSR 353 Expert Group and specification download:

<https://json-processing-spec.java.net/>

# How might JSON be used by PESC and what are the advantages

The most useful approach would be to allow instance documents created and validated in XML to then be converted to JSON for transport to a site accepting JSON. Since the product of a standards workgroup is an XML Schema, this is the only way to assure that the JSON output was PESC compliant. If we can substantiate that rules for conversion maintain the XML standards structure in JSON then we can certify this use.

# Issues of converting XML to JSON

Issues:

1. If XML has the same element tag in a complex type and the element is not consecutive and must be in sequence then this cannot be expressed in JSON

<x>

<a>test</a>

<b>test2</b>

<a>test3</a>

</x>

{“x”:{“a”:”test1”,”b”:”test2”,”a”:”test3”}} illegal

{“x”:{“a”:[“test1”,”test3”],”b”:”test2”}} sequence not preserved

1. Handling of attributes:

Use @ to indicate attribute

1. Handling of namespaces:

Put in namespace as part of name

1. Handling optional tags:
2. Handling empty tags: <a/>

# Other XML organizations using JSON

**ISO**

I could not find anything on the ISO site that even mentioned JSON. I did Google searches for '“ISO" and “JSON"' and found a number of articles on how to render/encode various ISO standards (3166 country codes, dates, etc), or how to handle difficulties with rendering between JSON and an ISO standard represented as XML.  One item of note from the ISO search is this item in an IBM document I came across regarding what to use for DB/2 on the zOS operating system (XML or JSON):

"XML is suitable for data exchange or sharing between independent entities, systems, or applications, particularly where the domain is regulated. It allows third parties to define portions of data structures independently (for example, banking or insurance).

JSON is suitable for use for data exchange or sharing within an application. It's typically used with human interfaces and mobile applications, making it straight-forward to pass data structures back and forth.

**OASIS**

There was a bit more information regarding OASIS and JSON, but not all that much indicated (at least to me) major development.  
  
The following link lists OASIS Technical Committees that either currently feature JSON and/or REST in their charters, or are discussing REST or JSON:  
  
<https://www.oasis-open.org/resources/topics/rest-json>  
  
OASIS has also recently approved version 4.0 of the Open Data Protocol (OData) and the OData JSON Format. A press release on MarketWatch is here:  
  
<http://www.marketwatch.com/story/oasis-approves-odata-40-standards-for-an-open-programmable-web-2014-03-17>  
  
Not much, if anything, related to XML standards. Just discussion of OData in mobile and web applications.

# Languages that support JSON parsing

* C  (jason-parser)
* awk (json.awk)
* C++ (a bunch, including JSONKit, JSON++ and libjson)
* C# (JSON for .net, JSONSharp, Manatee Json)
* Javascript (JSON, kson2.js, clarinet)
* Java (JSON Tools, google-gson, Argo, SOJO, XStream, Json-lib, jjson)
* Objective C (JSONKit, NSJSONSerialization, json-framework, ObjFW)
* Perl (CPAN, perl-JSON-SL)
* PHP (native in 5.2, Services\_JSON, json)
* PL/SQL (pljson, Librarie-JSON)
* Python (standard library, simplejson, pyson, ultraison)
* Ruby (built-in)
* Visual Basic (VB-JSON, PW.JSON)

# How do SISs support JSON exchanges? Gideon

# Tools available for creating JSON from XML

From David, **CAM** is one answer to the question.

He said that “**CAM tools now support JSON out the box.** CAM is using the JAXB - XML2JSON library.  The style of this is pretty self-explanatory although there might be a formal method specification it is following.

Round trip XML -> JSON -> XML works with this library and approach - no issues (e.g. namespaces).

To see this in action - try these two links for same identical content rendering:  
XML - <http://verifyxml.org/restOpenXDXPharmacy/resources/XML?zip=60624,60619&vaccType=1>  
JSON - <http://verifyxml.org/restOpenXDXPharmacy/resources/JSON?zip=60624,60619&vaccType=1>”.

David’s recommend is that no action needed to support JSON, he said folks can take the XML schema and just render to and from JSON quite happily if they have JSON specific tooling that needs that format.

The Java JAXB library fully supports that, and many other languages also have similar capabilities built-in.

Besides that I did some search for the assignment on “Is there a tool **to generate a JSON schema from an XML schema through JAVA?”**

It seems thatConverting XML to JSON is quite easy and can be done various ways:

<http://answers.oreilly.com/topic/278-how-to-convert-xml-to-json-in-java>

<http://www.json.org/javadoc/org/json/XML.html#toJSONObject%28java.lang.String%29>

There are other links for converting from XML to JSON:

<http://www.bramstein.com/projects/xsltjson/>

<http://code.google.com/p/xml2json-xslt/>

<http://json-lib.sourceforge.net/index.html>

Also, **Altova's XML Spy tool will do this.** They have a 30-day free trial that can be used to try it out: <http://www.altova.com/xmlspy.html>.