

General

API Data Exchange: XML vs. JSON



by Rob Zazueta

January 23, 2014

● General



"How Do You Spell API?" is a monthly blog series where Mashery experts break down complex API-related topics into language we can all understand.

API designers these days tend to land on one of two formats for exchanging data between their servers and client developers – XML or JSON. Though a number of different formats for data have been designed and promoted over the years, XML's built in validation properties and JSON's agility have helped both formats emerge as leaders in the API space.

XML – Ideal for Highly Structured Information

If you're familiar with HTML, you'll recognize XML – which is short for "eXtensible Markup Language". When combined with an XML Schema Definition (XSD), which defines how an XML document must be structured, the data exchanged between two systems can be validated for consistency and completion on the fly, offloading that responsibility from the code that will process that data.

Modeling a basic product object in XML would look something similar to this:

```
<product>
  <id>15</id>
  <name>Widgets</name>
  <description>These widgets are the finest widgets ever made by anyone.</description>
  <options type="color">
    <item>Purple</item>
    <item>Green</item>
    <item>Orange</item>
  </options>
</product>
```

Parameter labels are found within the tags enclosed in angle brackets, such as the top level <product> item. Attributes can also be assigned within those tags, such as the "type" property set to "color" within the <options> tag.

Due in no small part to the popularity of HTML, XML emerged as a front runner to represent data exchanged via APIs early on. The SOAP and XML-RPC protocols both rely heavily on XML not only to provide the data in their responses, but also to accept the requests. SOAP in particular gained wide acceptance in critical transactional systems, such as those found in financial institutions and large enterprises, because of its strongly typed data and strictly enforced structure.

But this extra validation doesn't come cheaply. Parsing an XML document and validating it against an XSD can take a fair amount of memory and computing power. The addition of tags and attributes lends extra weight to the data payload, which can significantly affect the performance of applications in constrained environments like mobile and embedded systems.

JSON – Lean and Mean

Javascript Object Notation (JSON), emerged as a standard for easily exchanging Javascript object data between systems. Modern Javascript is designed to natively read that data and deserialize it into objects, making it available to the rest of the code running in the system. As computing power increased alongside improved network bandwidth, Javascript evolved into a mature and powerful language running entirely within the web browser. With Javascript as a client, many API producers began returning data in its native format, avoiding the need for black box XML code libraries and the bloat that often comes with them.

JSON's simplicity has made it a favored data exchange format for several other agile languages as well, especially the Ruby community. JSON is easier to parse than XML and its structure is much lighter. Take, for example, the product

object we've already discussed and see how it could be rendered in JSON:

```
{
  "product": {
    "id": 15,
    "name": "Widgets",
    "description": "These widgets are the finest widgets ever made by anyone.",
    "options": [
      {
        "type": "color",
        "items": [
          "Purple",
          "Green",
          "Orange"
        ]
      }
    ]
  }
}
```

There are fewer characters to be passed on the wire, which can save gigabytes of transfer when applied to a high-traffic API.

JSON's biggest weakness is its lack of defined data structures. Proponents of XML have created a series of data formats that can be used to easily exchange and validate data across disparate systems. The Schema.org site, supported by several large companies including Google, Yahoo and Microsoft, acts as a repository for many of these definitions.

JSON, however, only defines simple variables, arrays and hashes – arrays that use strings instead of numbers as the index – and little else. As a result, API producers have frequently developed their own JSON response formats in the absence of well-defined standards. In the last couple of years, some standards have begun to emerge such as HAL, Siren and JSON-LD – [which recently gained acceptance as a W3C Recommendation](#). But it will be some time before the dust settles and API producers change their responses to conform to these standards.

So, Which One is Right?

RESTful API's should be designed to be fast, reliable and easy to use. JSON is becoming the data exchange format of choice because it aligns so well with those goals. But, until standards shake out that allow developers to use more generalized clients to parse JSON data and provide strict type and format validation, XML will likely be the format of choice for API developers most concerned about providing a rigid data structure. We may well see the JSON standards shake out in the next year or two, which means XML may soon be consigned the same fate as floppy disks and punch cards.



« Previous Story

When Do You Need to Involve Your IT Organization?

Next Story »

Using Big Data Analytics To Unlock the Mysteries of the Universe

Related Posts



TUCON 2013 Live
Blog: Kickoff

Leave a Comment

Name*

Email*

Website

Add Your Comment

ADD COMMENT

 @TIBCO

Wednesday Jul 18 • 7:17pm

Incredible shot - [@aliACTIONjackso](#) is pure speed! [#TIBCOfast](#) [#ToTheTop](#) [@teamTIBCO](#) <https://t.co/ts295QsTRE>

Wednesday Jul 18 • 3:14pm

Can [#blockchain](#) improve [#IoT](#)? In this piece for [@ForbesTech](#), TIBCO CTO [@NelsonPetracek](#) discusses how these technolo... <https://t.co/5La4gaH2ou>

Follow Us »

Follow Us



TIBCO Cloud™

Simplify the way you capture data, connect applications, and turn decisions into actions.

[Free Trial](#)

Subscribe to The TIBCO Blog updates

-
- ☒ Integration
 - ☒ Analytics
 - ☒ Event Processing
 - ☒ Customer Success
 - ☒ Tips & Tricks
 - ☒ Thought Leadership

Follow us



Copyright © 2018 TIBCO Software Inc. All rights reserved.

[Privacy Policy](#) | [Terms of Use](#)

[Site Map](#) | [Trademarks](#)