JSON Redux AKA RFC7159

The IETF has just revised its JSON spec; the new version is [RFC7159](#) — that link is to the IETF's traditional line-printer format, I've parked an HTML version at [rfc7159.net](#) for people who want to actually read the thing not just link to it. *[Disclosure: I edited RFC7159.]*

Highlights · RFC7159 cleans up some ambiguities and inconsistencies in various JSON definitions, none of which caused any real-world pain. [¶](#)

More important, it captures industry experience about stupid things you can do in your JSON that are allowed by the spec but will cause problems in practice. If you're interested, I recommend opening up [the HTML version](#) and searching forward for the string "interop". There are 17 occurrences. If you're generating JSON — something a lot of us do all the time — and make sure you avoid the mistakes highlighted in those 17 places, you're very unlikely to cause pain or breakage in software that's receiving it.

History · JSON was invented by [Douglas Crockford](#). When I first came across it in the early 2000s, I poked around and ended up at Crockford's [JSON.org](#), which remains an excellent example of how to do a technical tutorial right; short, clear, and with no BS. [¶](#)

In 2005 or so the IETF noticed that gobs and gobs of JSON were being sent back and forth with HTTP and the rules say that when you do that you should have a "Content-type" header with a registered Media Type. There wasn't one, so Crockford pitched in and we got [RFC 4627](#), which was probably what most people thought of as the official definition of JSON.

Since the "JS" in JSON stands for JavaScript, and the official spec for that is done in ECMA (where it's called "ECMAScript"), it shouldn't be surprising that another description of JSON turned up in section 15.12 of the [ECMAScript Version 5.1 spec](#).

Which brings us to 2013. There was considerable grumbling in the IETF over JSON specs and in particular RFC4627. First of all, it was "Informational" rather than "Standards track" which meant there were bureaucratic problems referring to it in certain contexts. Secondly, it and the ECMA version were inconsistent in that 4627 required a JSON text to be either an object or a list, whereas the ECMA version was fine with just `"42"` or `true`. Finally, 4627 allowed a few things, like duplicate keys in objects and broken Unicode strings, that everyone agrees are bad practices.

So the IETF spun up a [JSON Working Group](#) in 2013 with the objective of revising 4627 to fix these problems. Douglas Crockford signed up to edit the revision, but couldn't stand the chaos and noise of the IETF process and stepped away. Since I'd been making suggestions on the mailing list and [have experience editing markup-language specifications](#), they asked me if I'd mind finishing up the editing process. I didn't think it would be much work (I was right, it wasn't) and agreed.

Then we had a completely ridiculous standards-organization kitten-fight; someone told the ECMA working group that the IETF had gone crazy and was going to rewrite JSON with no regard for compatibility and break the whole Internet and something had to be done urgently about this terrible situation. Apparently nobody bothered to read the IETF Working Group's charter. So ECMA rushed out ECMA 404, yet another specification of JSON, on the double-quick hurry-up. It's *extremely* minimal, claims to specify only the syntax but not the semantics of JSON (I don't understand what they mean by those words). It doesn't address any of the gripes that were motivating the IETF revision.

It got pretty silly for a while, with demands that everyone officially recognize ECMA 404 as the only "normative" definition of JSON. But I think this little flurry of excitement left no permanent scars.

Anyhow, by my count, the Internet now has at least 5 different things that can be thought of as JSON specifications. Fortunately, everyone pretty well agrees on the right way to do JSON, so that's not a problem.

Comments on this fragment are closed.

Updated: 2014/03/06

March 05, 2014
· Technology (77 fragments)
· · Software (53 more)

By Tim Bray.

I am an employee
of Amazon.com, but
the opinions expressed here
are my own, and no other party
necessarily agrees with them.

A full disclosure of my
professional interests is
on the author page.