
JSON Encoding of Data Modeled with YANG

draft-ietf-netmod-yang-json-00

Ladislav Lhotka
<lhotka@nic.cz>

21 July 2014

Metadata Mapping

New section on mapping XML attributes to a “metadata object” in JSON.

Questions:

1. Is this draft an appropriate place for defining metadata encoding in JSON?
2. If so, does this mapping satisfy the needs?

I-JSON Compliance

I-JSON (*draft-ietf-json-i-json-02*) introduces additional restrictions in order to guarantee maximum interoperability for JSON-based communication protocols.

The present draft is almost compliant, except for

- `int64`, `uint64` and `decimal64` numbers,
- `binary` type.

Large Integers

RFC 7159 doesn't prescribe any internal representation of JSON numbers but experience shows that numbers outside the range of IEEE 754-2008 binary64 (double precision) often break interoperability.

"I-JSON sender MUST NOT expect a receiver to treat an integer whose absolute value is greater than 9007199254740991 (i.e., that is outside the range $\langle -2^{53} + 1, 2^{53} - 1 \rangle$) as an exact value.

For applications which require the exact interchange of numbers with greater magnitude or precision (one example would be 64-bit integers), it is RECOMMENDED to encode them in JSON string values."

Question: Should we encode `int64`, `uint64` and `decimal64` values as strings?

Binary Data

“When it is required that an I-JSON protocol element contain arbitrary binary data, it is RECOMMENDED that this data be encoded in a string value in *base64url* (RFC 4868, section 5).”

YANG binary type uses *base64*. It would be problematic to have different encodings for XML and JSON, and URL-safe encoding is probably not needed.

Proposal: Stick to *base64*.