Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2014

T.B. Bray, Ed.
Google, Inc.
July 11, 2013

The I-JSON Message Format
draft-bray-i-json-00

Abstract

I-JSON is a restricted profile of JSON designed to maximize interoperability and increase confidence that software can process it successfully with predictable results.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RFC4627bis specifies a data format called JSON which has come to be widely used in Internet protocols. For historical reasons, that RFC allows the use of language idioms and text encoding patterns which are likely to lead to interoperability problems and software breakage, particularly when a program receiving JSON data uses automated software to map it into native programming-language structures or database records.

This document specifies I-JSON, short for "Internet JSON". I-JSON Messages are also JSON texts per RFC4627bis, but with improved interoperability and lower risk of breakage in receiving software.

1.1. Terminology

The terms "object", "member", "array", "number", "name", and "string" in this document are to be interpreted as described in RFC4627bis.

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. I-JSON Messages

An I-JSON Message is a "JSON text" as defined by RFC4626bis and therefore MUST be either an object or an array.

For maximum flexibility, an I-JSON Message SHOULD be an object. This allows self-identification (see below) and also allows protocol

designers to add new data items to messages, should that become necessary, without breaking existing deployments. In other words, it makes a Must-Ignore policy possible.

2.1. Self-identification

If an I-JSON Message is an object, it MAY self-identify by including a member whose name is "urn:ietf:i-json" and whose value is an object, which MUST be the first member of the top-level object. This specification does not constrain the content of the object; it might be useful for further profiling in future specifications.

When an I-JSON message is an HTTP request or response body, identified with the Internet Media Type "application/json" or alternately of the "+json" form as described in RFC 6838 [RFC6838], a media-type parameter MAY be included of the form "profile=i-json"; so the whole media type would be "application/json; profile=i-json" or "application/XXX+json; profile=i-json".

2.2. Encoding and Characters

I-JSON Messages MUST be encoded using UTF-8.

String values of object members in I-JSON Messages MUST NOT include code points which identify Surrogates or Noncharacters (Unicode section 2.4), and SHOULD NOT include code points which identify Compatibility Characters (Unicode section 2.3) or Control characters (Unicode section 2.4).

This applies both to characters encoded directly in UTF-8 and to those which are escaped; thus, "\uDDDD" is illegal.

2.3. Numbers

Number values of object members in I-JSON Messages must be exactly representable as IEEE 754:2008 64-bit binary floating point numbers.

Numbers of greater length are likely to cause breakage when the receiving program is in a statically-typed language or in JavaScript. For applications such as cryptography, where much larger numbers are reasonably required, it is RECOMMENDED to transmit large numbers as strings. This requires that the receiving program understand the intended semantic of the member.

2.4. Object member names

Objects in I-JSON Messages MUST NOT have members with duplicate names.

3. Software Behavior

When software reads data which it expects to be an I-JSON message, but the data violates one of the MUST constraints in the previous section (for example, contains an object with a duplicate key, or a UTF-8 encoding error), that software MUST NOT trust nor act on the content of the message.

Designers of protocols which use I-JSON messages SHOULD provide a way, in this case, for the receiver of the erroneous data to signal the problem to the sender.

4. Acknowledgements

I-JSON is entirely dependent on the design of JSON, largely due to Douglas Crockford. The specifics were strongly influenced by the contributors to the design of RFC4627bis on the IETF JSON Working Group.

5. IANA Considerations

IANA will need to define a new entry in the "urn:ietf" namespace, "urn:ietf:i-json", to support I-JSON self-identification.

IANA will need to register the "profile" parameter and value "i-json" for the "application/json" media type, and take steps to allow the use of this parameter on media types of the form "application/XXX+json".

6. Security Considerations

All the security considerations which apply to JSON (see RFC4627bis) apply to I-JSON. There are no additional security considerations specific to I-JSON.

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

Author's Address

Tim Bray (editor)
Google, Inc.

Email: tbray@textuality.com
URI: https://www.tbray.org/