Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 23, 2018

G. Luff

H. Andrews, Ed.
Cloudflare, Inc.

November 19, 2017

Relative JSON Pointers
draft-handrews-relative-json-pointer-00

Abstract

JSON Pointer is a syntax for specifying locations in a JSON document, starting from the document root. This document defines an extension to the JSON Pointer syntax, allowing relative locations from within the document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 23, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Conventions and Terminology 2
3. Syntax 2
4. Evaluation 3
5. JSON String Representation 4
5.1. Examples 4
6. Non-use in URI Fragment Identifiers 5
7. Error Handling 5
8. Relationship to JSON Pointer 5
9. Acknowledgements 5
10. Security Considerations 5
11. References 6
11.1. Normative References 6
11.2. Informative References 6
Appendix A. ChangeLog 7
Authors' Addresses 7

1. Introduction

JSON Pointer (RFC 6901 [RFC6901]) is a syntax for specifying locations in a JSON document, starting from the document root. This document defines a related syntax allowing identification of relative locations from within the document.

2. Conventions and Terminology

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].

3. Syntax

A Relative JSON Pointer is a Unicode string (see RFC 4627, Section 3 [RFC4627]), comprising a positive integer, followed by either a '#' (%x23) character or a JSON Pointer (RFC 6901 [RFC6901]).

The separation between the integer prefix and the JSON Pointer will always be unambiguous, because a JSON Pointer must be either zero-length or start with a '/' (%x2F). Similarly, a JSON Pointer will never be ambiguous with the '#'.

The ABNF syntax of a Relative JSON Pointer is:

```
relative-json-pointer =  positive-integer &lt;json-pointer&gt;
relative-json-pointer =/ positive-integer "#"
positive-integer      =  %x30 / %x31-39 *( %x30-39 )
        ; "0", or digits without a leading "0"
```

where <json-pointer> follows the production defined in RFC 6901, Section 3 [RFC6901] ("Syntax").

4. Evaluation

Evaluation of a Relative JSON Pointer begins with a reference to a value within a JSON document, and completes with either a value within that document, a string corresponding to an object member, or integer value representing an array index.

Evaluation begins by processing the positive-integer prefix. This can be found by taking the longest continuous sequence of decimal digits available, starting from the beginning of the string, taking the decimal numerical value. If this value is more than zero, then the following steps are repeated that number of times:

- If the current referenced value is the root of the document, then evaluation fails (see below).
- If the referenced value is an item within an array, then the new referenced value is that array.
- If the referenced value is an object member within an object, then the new referenced value is that object.

If the remainder of the Relative JSON Pointer is a JSON Pointer, then evaluation proceeds as per RFC 6901, Section 4 [RFC6901] ("Evaluation"), with the modification that the initial reference being used is the reference currently being held (which may not be root of the document).

Otherwise (when the remainder of the Relative JSON Pointer is the character '#'), the final result is determined as follows:

- If the current referenced value is the root of the document, then evaluation fails (see below).
- If the referenced value is an item within an array, then the final evaluation result is the value's index position within the array.

If the referenced value is an object member within an object, then the new referenced value is the corresponding member name.

5. JSON String Representation

The concerns surrounding JSON String representation of a Relative JSON Pointer are identical to those laid out in RFC 6901, Section 5 [RFC6901].

5.1. Examples

For example, given the JSON document:

```
{
   "foo": ["bar", "baz"],
   "highly": {
      "nested": {
         "objects": true
      }
   }
}
```

Starting from the value "baz" (inside "foo"), the following JSON strings evaluate to the accompanying values:

"0"	"baz"
"1/0"	"bar"
"2/highly/nested/objects"	true
"0#"	1
"1#"	"foo"

Starting from the value {"objects":true} (corresponding to the member key "nested"), the following JSON strings evaluate to the accompanying values:

"0/objects"	true
"1/nested/objects"	true
"2/foo/0"	"bar"
"0#"	"nested"
"1#"	"highly"

6. Non-use in URI Fragment Identifiers

Unlike a JSON Pointer, a Relative JSON Pointer can not be used in a URI fragment identifier. Such fragments specify exact positions within a document, and therefore Relative JSON Pointers are not suitable.

7. Error Handling

In the event of an error condition, evaluation of the JSON Pointer fails to complete.

Evaluation may fail due to invalid syntax, or referencing a non-existent value. This specification does not define how errors are handled. An application of JSON Relative Pointer SHOULD specify the impact and handling of each type of error.

8. Relationship to JSON Pointer

Relative JSON Pointers are intended as a companion to JSON Pointers. Applications MUST specify the use of each syntax separately. Defining either JSON Pointer or Relative JSON Pointer as an acceptable syntax does not imply that the other syntax is also acceptable.

9. Acknowledgements

The language and structure of this specification are based heavily on [RFC6901], sometimes quoting it outright.

This draft remains primarily as written and published by Geraint Luff, with only minor subsequent alterations under new editorship.

10. Security Considerations

Evaluation of a given Relative JSON Pointer is not guaranteed to reference an actual JSON value. Applications using Relative JSON Pointer should anticipate this situation by defining how a pointer that does not resolve ought to be handled.

As part of processing, a composite data structure may be assembled from multiple JSON documents (in part or in full). In such cases, applications SHOULD ensure that a Relative JSON Pointer does not evaluate to a value outside the document for which is was written.

Note that JSON pointers can contain the NUL (Unicode U+0000) character. Care is needed not to misinterpret this character in programming languages that use NUL to mark the end of a string.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6901] Bryan, P., Ed., Zyp, K., and M. Nottingham, Ed., "JavaScript Object Notation (JSON) Pointer", RFC 6901, DOI 10.17487/RFC6901, April 2013, <https://www.rfc-editor.org/info/rfc6901>.

11.2. Informative References

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <https://www.rfc-editor.org/info/rfc4627>.

Appendix A. ChangeLog

[[CREF1: This section to be removed before leaving Internet-Draft status.]]

draft-handrews-relative-json-pointer-00

* Revived draft with identical wording and structure.

* Clarified how to use alongside JSON Pointer.

draft-luff-relative-json-pointer-00

* Initial draft.

Authors' Addresses

Geraint Luff
Cambridge
UK

EMail: luffgd@gmail.com

Henry Andrews (editor)
Cloudflare, Inc.
San Francisco, CA
USA

EMail: henry@cloudflare.com