

Internet Engineering Task Force	P. Bryan, Ed.
Internet-Draft	ForgeRock US, Inc.
Intended status: Informational	K. Zyp
Expires: April 23, 2012	SitePen (USA)
	October 21, 2011

JSON Pointer
draft-pbryan-zyp-json-pointer-02

Abstract

JSON Pointer defines a syntax for identifying a specific value within a JSON 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 <http://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 April 23, 2012.

Copyright Notice

Copyright (c) 2011 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 (<http://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. [Conventions](#)
- *3. [Syntax](#)
- *4. [Evaluation](#)
- *5. [Error Handling](#)
- *6. [IANA Considerations](#)
- *7. [Security Considerations](#)
- *8. [References](#)

*Appendix A. [Examples](#)

*[Authors' Addresses](#)

1. Introduction

This specification defines JSON Pointer, a syntax for identifying a specific value within a [JSON \[RFC4627\]](#) text document. The syntax is intended to be expressed in a JSON string value or the URI fragment identifier of a resource that has a JSON representation.

2. Conventions

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 [\[RFC2119\]](#).

This specification expresses syntax rules using [ABNF \[RFC5234\]](#) (Augmented Backus-Naur Form) notation. Additionally, it uses the following rules from [URI syntax \[RFC3986\]](#): "unreserved" and "pct-encoded".

3. Syntax

A JSON Pointer is a sequence of zero or more reference tokens, each prefixed by a "/" (%x2F) character. Each reference token is a sequence of unreserved and/or percent-encoded characters, per [\[RFC3986\]](#).

```
json-pointer = *( "/" reference-token )
reference-token = *( unreserved / pct-encoded )
```

Characters in reference tokens that are not unreserved SHOULD be percent-encoded, per [\[RFC3986\]](#), and MUST be so encoded as "%2F" if the character is "/" to avoid being interpreted as a reference token prefix.

It is an error condition if a JSON Pointer does not conform to this syntax.

4. Evaluation

Evaluation begins with a reference to the root value of a JSON text document, and completes with a reference to some value within the same document. Each token in the JSON Pointer sequence is evaluated, changing the referenced value according to the following scheme: If the currently referenced value is a JSON object, the new referenced value is the object member with the name identified by the token. If the currently referenced value is a JSON array, the token MUST contain an unsigned base-10 integer value, and the new referenced value is the array element with the zero-based index identified by the token. If a reference token is being evaluated against a concrete JSON text document, the implementation MAY evaluate each token against a concrete value, and terminate evaluation with an error condition if a reference token fails to resolve to a concrete value. If a concrete JSON object contains a referenced member with a non-unique name, which of the matching members is resolved is undefined.

5. Error Handling

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

6. IANA Considerations

This draft includes no request to IANA.

7. Security Considerations

A pointer is not guaranteed to resolve to a JSON value. Implementations of JSON Pointer should be aware of this and take appropriate precautions.

8. References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels" , BCP 14, RFC 2119, March 1997.
[RFC3986]	Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax" , STD 66, RFC 3986, January 2005.
[RFC4627]	Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)" , RFC 4627, July 2006.
[RFC5234]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF" , STD 68, RFC 5234, January 2008.

Appendix A. Examples

The following examples illustrate the use of JSON pointers in URI fragments for a JSON text document located at `http://example.com/example.json`, with the following value:

```
{
  "foo": {
    "bar": [ "element0", "element1" ],
    "inner object": {
      "baz": "qux"
    }
  }
}
```

`http://example.com/example.json#`

Resolves to the object value at the root of the JSON text document.

`http://example.com/example.json#/foo`

Resolves to the object value of the "foo" member in the root object.

`http://example.com/example.json#/foo/inner%20object`

Resolves to the object value of the "inner object" member in the "foo" object value in the root object.

`http://example.com/example.json#/foo/inner%20object/baz`

Resolves to the string value "qux", which is the value of the "baz" member in the "inner object" member in the "foo" member in the root object.

`http://example.com/example.json#/foo/bar/0`

Resolves to the string value "element0", which is the first value in the "bar" array in the "foo" member in the root object.

Authors' Addresses

Paul C. Bryan editor Bryan ForgeRock US, Inc. 201 NE Park Plaza
Drive Suite 196 Vancouver, WA 98684 USA Phone: +1 604 783 1481
EMail: paul.bryan@forgerock.com

Kris Zyp Zyp SitePen (USA)
530 Lytton Avenue Palo Alto, CA 94301 USA Phone: +1 650 968 8787
EMail: kris@sitepen.com