Applications Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 22, 2016

C. Bormann
Universitaet Bremen TZI
P. van der Stok
Consultant
March 21, 2016

CBOR Merge Patch
draft-bormann-appsawg-cbor-merge-patch-00

Abstract

This specification defines the CBOR merge patch format and processing rules, as an analog to the JSON merge patch format defined in RFC 7396. The merge patch format is primarily intended for use with the HTTP PATCH method and potential related CoAP methods as a means of describing a set of modifications to a target resource's content.

This specification also defines how a JSON merge patch document is applied to a CBOR data item and vice versa.

Table of Contents

1. Introduction

The Concise Binary Object Representation (CBOR, [RFC7049]) defines a binary representation for data that can be described using an extension of the JSON data model [RFC7159]. The JSON merge-patch [RFC7396] format was designed to represent updates ("patches") to data that is structured according to the JSON data model.

This specification defines the CBOR merge patch document format, processing rules, and associated MIME media type identifier. The merge patch format is primarily intended for use as a means of describing a set of modifications to a target resource's content, with the HTTP PATCH method [RFC5789] or with potential PATCH-like CoAP [RFC7252] methods to be defined [I-D-vanderstok-core-etch].

A CBOR merge patch document describes changes to be made to a target CBOR document using a syntax that closely mimics the document being modified. Recipients of a merge patch document determine the exact set of changes being requested by comparing the content of the provided patch against the current content of the target document. If the provided merge patch contains members that do not appear within the target, those members are added. If the target does contain the member, the value is replaced. Null values in the merge patch are given special meaning to indicate the removal of existing values in the target.

For example, given the following original CBOR document (as with all following examples, the CBOR data is shown in CBOR diagnostic notation):

```
{
  "a": h'4711',
  3: {
    "d": 1(1454280297),
    "f": "g"
  }
}
```

Changing the value of "a" and removing "f" can be achieved by sending:

```
PATCH /target HTTP/1.1
Host: example.org
Content-Type: application/merge-patch+cbor

{
  "a": "now is text",
  3: {
    "f": null
  }
}
```

When applied to the target resource, the value of the "a" member is replaced with "now is text" and "f" is removed, leaving the remaining content untouched.

This design means that merge patch documents are suitable for describing modifications to CBOR documents that primarily use maps for their structure and do not make use of explicit null values. The merge patch format is not appropriate for all CBOR structures.

> Discussion: CBOR has more simple types than JSON. We could define another simple type (beyond null) that is used for removing map entries. The current proposal does not do this as map entries with a null value are equivalent to not having the map entry in many CBOR structures.

This design also means that applying a CBOR merge patch is an idempotent operation: Applying the same patch again to the result of applying it first yields exactly the same result. This property may be useful in the context of a potential CoAP idempotent PATCH method (iPATCH in [I-D-vanderstok-core-etch]).

2. Processing Merge Patch Documents

CBOR merge patch documents describe, by example, a set of changes that are to be made to a target resource. Recipients of merge patch documents are responsible for comparing the merge patch with the current content of the target resource to determine the specific set of change operations to be applied to the target.

To apply the merge patch document to a target resource, the system realizes the effect of the following function, described in pseudocode. For this description, the function is called MergePatch, and it takes two arguments: the target resource document and the merge patch document. The Target argument can be any CBOR value or undefined. The Patch argument can be any CBOR value.

```
define MergePatch(Target, Patch):
  if Patch is a map:
    if Target is not a map:
      Target = {} # Ignore the contents and set it to an empty Map
    for each Name/Value pair in Patch:
      if Value is null:
        if Name exists in Target:
          remove the Name/Value pair from Target
      else:
        Target[Name] = MergePatch(Target[Name], Value)
    return Target
  else:
    return Patch
```

There are a few things to note about the function. If the patch is anything other than a map, the result will always be to replace the entire target with the entire patch. Also, it is not possible to patch part of a target that is not a map, such as to replace just some of the values in an array.

The MergePatch operation is defined to operate at the level of data items, not at the level of binary representation. There is no expectation that the MergePatch operation will preserve features at the textual-representation level such as member ordering or number precision beyond what is available in the target's implementation, and so forth.

3. Example

Given the following example CBOR document:

```
{
  "title": "Goodbye!",
  "author" : {
    "givenName" : "John",
    "familyName" : "Doe"
  },
  "tags": ["example", "sample"],
  "content": "This will be unchanged"
}
```

A user agent wishing to change the value of the "title" member from "Goodbye!" to the value "Hello!", add a new "phoneNumber" member, remove the "familyName" member from the "author" map, and replace the "tags" array so that it doesn't include the word "sample" would send the following request:

```
PATCH /my/resource HTTP/1.1
Host: example.org
Content-Type: application/merge-patch+cbor

{
  "title": "Hello!",
  "phoneNumber": "+01-123-456-7890",
  "author": {
    "familyName": null
  },
  "tags": ["example"]
}
```

The resulting CBOR document would be:

```
{
  "title": "Hello!",
  "author" : {
    "givenName" : "John"
  },
  "tags": ["example"],
  "content": "This will be unchanged",
  "phoneNumber": "+01-123-456-7890"
}
```

4. Interoperation of JSON merge patch and CBOR merge patch

A CBOR merge patch document is applied to a JSON document by first converting it into a JSON document using the rules in Section 4.1 of [RFC7049] and then applying the result as a JSON merge-patch document. This only works very well if the conversion from CBOR values in the CBOR merge patch document to JSON values results in a

useful JSON merge patch document. (This is trivially the case if all the data in the CBOR merge patch document conform to the unextended JSON data model. The effects of the conversions may also be desired, e.g. to patch a JSON map entry with the key "1" by supplying an integer 1 as the key in the patch, or to supply a base64url value in the form of an unencoded binary string.)

A JSON merge patch document is applied to a CBOR data items by first converting it into a CBOR data item using the rules in Section 4.2 of [RFC7049] and then applying the result as a CBOR merge patch document. The conversion always leads to a valid CBOR merge patch document, but the set of such documents that can be generated from JSON of course cannot create all possible outcomes that may be desired when patching a CBOR data item.

5. IANA Considerations

The Internet media type [RFC6838] for CBOR merge patch is application/merge-patch+cbor.

Type name: application

Subtype name: merge-patch+cbor

Required parameters: None

Optional parameters: None

Encoding considerations: Resources that use the "application/merge-patch+cbor" media type are required to conform to the "application/cbor" media type and are therefore subject to the same encoding considerations specified in Section 7.3 of [RFC7049].

Security considerations: As defined in this specification

Published specification: This specification.

Applications that use this media type: None currently known.

Additional information:

Magic number(s): N/A

File extension(s): N/A

Macintosh file type code(s): N/A

Person & email address to contact for further information: IESG

Intended usage: COMMON

Restrictions on usage: None

Author: Carsten Bormann <cabo@tzi.org>

Change controller: IESG

6. Security Considerations

The security considerations of [RFC7049] and [RFC7396] apply.

7. References

7.1. Normative References

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <http://www.rfc-editor.org/info/rfc7049>.

7.2. Informative References

[I-D-vanderstok-core-etch] van der Stok, P., Bormann, C., and A. Sehgal, "Patch and Fetch Methods for Constrained Application Protocol (CoAP)", March 2016.

[RFC5789] Dusseault, L. and J. Snell, "PATCH Method for HTTP", RFC 5789, DOI 10.17487/RFC5789, March 2010, <http://www.rfc-editor.org/info/rfc5789>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[RFC7396] Hoffman, P. and J. Snell, "JSON Merge Patch", RFC 7396, DOI 10.17487/RFC7396, October 2014, <http://www.rfc-editor.org/info/rfc7396>.

Appendix A. Example Test Cases

```
ORIGINAL        PATCH            RESULT
-----------------------------------------
{"a":"b"}       {"a":"c"}       {"a":"c"}

{"a":"b"}       {"b":"c"}       {"a":"b",
                                 "b":"c"}

{"a":"b"}       {"a":null}      {}

{"a":"b",       {"a":null}      {"b":"c"}
 "b":"c"}

{"a":["b"]}     {"a":"c"}       {"a":"c"}

{"a":"c"}       {"a":["b"]}     {"a":["b"]}

{"a": {         {"a": {         {"a": {
  "b": "c"}       "b": "d",       "b": "d"
}                 "c": null}      }
                }               }

{"a": [         {"a": [1]}      {"a": [1]}
  {"b":"c"}
 ]
}

["a","b"]       ["c","d"]       ["c","d"]

{"a":"b"}       ["c"]           ["c"]

{"a":"foo"}     null            null

{"a":"foo"}     "bar"           "bar"

{"e":null}      {"a":1}         {"e":null,
                                 "a":1}

[1,2]           {"a":"b",       {"a":"b"}
                 "c":null}

{}              {"a":            {"a":
                 {"bb":           {"bb":
                  {"ccc":          {}}}
                   null}}}
```

Acknowledgments

Paul Hoffman and James Snell wrote RFC 7396, from which this document liberally borrows most of its text.

Authors' Addresses

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
EMail: cabo@tzi.org

Peter van der Stok
Consultant

EMail: consultancy@vanderstok.org