

CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

K. Li
Alibaba Group
A. Rahman
InterDigital
C. Bormann, Ed.
Universitaet Bremen TZI
July 06, 2015

Representing CoRE Formats in JSON and CBOR
draft-ietf-core-links-json-03

Abstract

JavaScript Object Notation, JSON (RFC7159) is a text-based data format which is popular for Web based data exchange. Concise Binary Object Representation, CBOR (RFC7049) is a binary data format which has been optimized for data exchange for the Internet of Things (IoT). For many IoT scenarios, CBOR formats will be preferred since it can help decrease transmission payload sizes as well as implementation code sizes compared to other data formats.

Web Linking (RFC5988) provides a way to represent links between Web resources as well as the relations expressed by them and attributes of such a link. In constrained networks, a collection of Web links can be exchanged in the CoRE link format (RFC6690). Outside of constrained environments, it may be useful to represent these collections of Web links in JSON, and similarly, inside constrained environments, in CBOR. This specification defines a common format for this.

Group Communication for the Constrained Application Protocol (RFC7390) defines a number of JSON formats for controlling communication between groups of nodes employing the Constrained Application Protocol (CoAP). In a similar vein, this specification defines CBOR variants of these formats.

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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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 | 3 |
| 1.1. Objectives | 3 |
| 1.2. Terminology | 4 |
| 2. Web Links in JSON and CBOR | 4 |
| 2.1. Background | 4 |
| 2.2. Information Model | 5 |
| 2.3. Additional Encoding Step for CBOR | 6 |
| 2.4. Examples | 7 |
| 2.4.1. Link Format to CBOR Example | 8 |
| 2.4.2. Link Format in JSON to CBOR Example | 9 |
| 3. Group Communication Management Objects in CBOR | 11 |
| 3.1. Background | 11 |
| 3.2. Information Model | 11 |
| 3.3. Mapping | 11 |
| 3.4. Group Communication Example | 12 |
| 4. IANA Considerations | 14 |
| 5. Security Considerations | 15 |
| 6. Acknowledgements | 15 |
| 7. References | 16 |
| 7.1. Normative References | 16 |
| 7.2. Informative References | 16 |
| Appendix A. Implementation | 17 |
| Authors' Addresses | 17 |

1. Introduction

Web Linking [RFC5988] provides a way to represent links between Web resources as well as the relations expressed by them and attributes of such a link. In constrained networks, a collection of Web links can be exchanged in the CoRE link format [RFC6690] to enable resource discovery, for instance by using the CoAP protocol [RFC7252].

The JavaScript Object Notation (JSON) [RFC7159] is a lightweight, text-based, language-independent data interchange format. JSON is popular in the Web development environment as it is easy for humans to read and write.

The Concise Binary Object Representation (CBOR) [RFC7049] is a binary data format which requires extremely small code size, allows very compact message representation, and provides extensibility without the need for version negotiation. CBOR is especially well suited for IoT environments because of these efficiencies.

When converting between a bespoke syntax such as that defined by [RFC6690] and JSON or CBOR, many small decisions have to be made. If left without guidance, it is likely that a number of slightly incompatible dialects will emerge. This specification defines a common approach for translating between the CoRE-specific bespoke formats, JSON and CBOR formats. Where applicable, mapping from other formats (e.g. CoRE Link Format) into JSON or CBOR is also described.

This specification defines a common format for representing CoRE Web Linking in JSON and CBOR, as well as the various JSON formats for controlling CoRE group communication [RFC7390], in CBOR.

Note that there is a separate question on how to represent Web links pointing out of JSON documents, as discussed e.g. in [MNOT11]. While there are good reasons to stay as compatible as possible to developments in this area, the present specification is solving a different problem.

1.1. Objectives

This specification has been designed based on the following objectives:

- o Canonical mapping
 - * lossless round-tripping with [RFC6690] and between JSON and CBOR
 - * but not trying for bit-preserving (DER-style) round-tripping

- o The simplest thing that could possibly work
 - * Do not cater for RFC 5988 complications caused by HTTP header character set issues [RFC2047]
- o Consider other work that has links in JSON, e.g.: JSON-LD, JSON-Reference [I-D.pbryan-zyp-json-ref]
 - * Do not introduce unmotivated differences

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

The term "byte" is used in its now customary sense as a synonym for "octet".

CoAP: Constrained Application Protocol [RFC7252]

CBOR: Concise Binary Object Representation [RFC7049]

CoRE: Constrained RESTful Environments, the field of work underlying [RFC6690], [RFC7049], [RFC7252], and [RFC7390]

IoT: Internet of Things

JSON: JavaScript Object Notation [RFC7159]

The objective of the JSON and CBOR mappings defined in this document is to contain information of the formats specified in [RFC5988] and [RFC6690]. This specification therefore uses the names of the ABNF productions used in those documents.

2. Web Links in JSON and CBOR

2.1. Background

Web Linking [RFC5988] provides a way to represent links between Web resources as well as the relations expressed by them and attributes of such a link. In constrained networks, a collection of Web links can be exchanged in the CoRE link format [RFC6690] to enable resource discovery, for instance by using the CoAP protocol [RFC7252] and in

conjunction with the CoRE resource directory [I-D.ietf-core-resource-directory].

2.2. Information Model

This section discusses the information model underlying the CORE Link Format payload.

An application/link-format document is a collection of web links ("link-value"), each of which is a collection of attributes ("link-param") applied to a "URI-Reference".

We straightforwardly map:

- o the outer collection to an array of links;
- o each link to a JSON object or CBOR map, mapping attribute names to attribute values.

In the object representing a "link-value", each target attribute or other parameter ("link-param") is represented by a JSON name/value pair (member). The name is a string representation of the parameter or attribute name (as in "parname"), the value is a string representation of the parameter or attribute value ("ptoken" or "quoted-string"). "quoted-string" productions are parsed (i.e, the outer quotes removed and the backslash constructions evaluated) as defined in [RFC6690] and its referenced documents, before placing them in JSON strings (where they may gain back additional decorations such as backslashes as defined in [RFC7159]).

If no attribute value ("ptoken" or "quoted-string") is present, the presence of the attribute name is indicated by using "true" as the value.

If a Link attribute ("parname") is present more than once in a "link-value", its values are then represented as a JSON array of JSON string values; this array becomes the value of the JSON name/value pair where the attribute name is the JSON name. Attributes occurring just once MUST NOT be represented as JSON arrays but MUST be directly represented as JSON strings. (Note that [RFC6690] has cut down on the use of repeated parameter names; they are still allowed by [RFC5988] though. No attempt has been made to decode the possibly space-separated values for rt=, if=, and rel= into JSON arrays.)

The URI-Reference is represented as a name/value pair with the name "href" and the URI-Reference as the value. (Rationale: This usage is consistent with the use of "href" as a query parameter for link-

format query filtering and with link-format reserving the link parameter "href" specifically for this use [RFC6690]).

The resulting structure can be represented in CDDL [I-D.greevenbosch-appsawg-cbor-cddl] as:

```
links = [* link]
link = {
  href: tstr      ; resource URI
  * tstr => tstr / true
}
```

Figure 1: CoRE Link Format Data Model

2.3. Additional Encoding Step for CBOR

The above specification for JSON could be used as is for the CBOR encoding as well. However, to further reduce message sizes, it is beneficial to perform an extra encoding step, and encode "href" and some commonly occurring attribute names as small integers.

The substitution is summarized below:

| name | encoded value |
|----------|---------------|
| href | 1 |
| rel | 2 |
| anchor | 3 |
| rev | 4 |
| hreflang | 5 |
| media | 6 |
| title | 7 |
| type | 8 |
| rt | 9 |
| if | 10 |
| sz | 11 |
| ct | 12 |
| obs | 13 |

Table 1: Integer Encoding of common attribute names

** TO DO: Is this the right list of attribute names? **

This list of substitutions is fixed by the present specification; no future expansion of the list is foreseen. "href" as well as all

attribute names in this list MUST be represented by their integer substitutions and MUST NOT use the attribute name in text form.

This leads to the following CDDL representation for the CBOR encoding:

```
links = [* link]
link = {
  href: tstr      ; resource URI
  * label => tstr / true
}
label = tstr / &(
  href: 1,    rel: 2,    anchor: 3,
  rev: 4,    hreflang: 5, media: 6,
  title: 7,   type: 8,    rt: 9,
  if: 10,    sz: 11,     ct: 12,
  obs: 13,
)
```

Figure 2: CoRE Link Format Data Model (CBOR)

2.4. Examples

```
</sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>;anchor="/sensors/temp"
;rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```

Figure 3: Example from page 15 of [RFC6690]

The link-format document in Figure 3 becomes (321 bytes):

```
"[{\"href\":\"/sensors\",\"ct\":\"40\",\"title\":\"Sensor
Index\"},{\"href\":\"/sensors/temp\",\"rt\":\"temperature-
c\",\"if\":\"sensor\"},{\"href\":\"/sensors/light\",\"rt\":\"light-
lux\",\"if\":\"sensor\"},{\"href\":\"http://www.example.com/sensors/
t123\",\"anchor\":\"/sensors/
temp\",\"rel\":\"describedby\"},{\"href\":\"/t\",\"anchor\":\"/sensors/
temp\",\"rel\":\"alternate\"}] "
```

(More examples to be added.)

2.4.1.1. Link Format to CBOR Example

This examples shows conversion from link format to CBOR format.

The link-format document in Figure 3 becomes (in CBOR diagnostic format):

```
[{1: "/sensors", 12: "40", 7: "Sensor Index"},
 {1: "/sensors/temp", 9: "temperature-c", 10: "sensor"},
 {1: "/sensors/light", 9: "light-lux", 10: "sensor"},
 {1: "http://www.example.com/sensors/t123", 3: "/sensors/temp",
  2: "describedby"},
 {1: "/t", 3: "/sensors/temp", 2: "alternate"}]
```

or, in hexadecimal (203 bytes):

```
85          # array(number of data items:5)
a3          # map(# data item pairs:3)
  01        # unsigned integer(value:1,"href")
  68        # text string(8 bytes)
    2f73656e736f7273    # "/sensors"
  0c        # unsigned integer(value:12,"ct")
  62        # text(2)
    3430          # "40"
  07        # unsigned integer(value:7,"title")
  6c        # text string(12 bytes)
    53656e736f7220496e646578    # "Sensor Index"
a3          # map(# data item pairs:3)
  01        # unsigned integer(value:1,"href")
  6d        # text string(13 bytes)
    2f73656e736f72732f74    # "/sensors/temp"
    656d70          # unsigned integer(value:9,"rt")
  09        # text string(13 bytes)
  6d        # "temperature-c"
    74656d70657261747572    # unsigned integer(value:10,"if")
    652d63          # text string(6 bytes)
  0a        # "sensor"
  66        # map(# data item pairs:3)
    73656e736f72    # unsigned integer(value:1,"href")
a3          # text string(14 bytes)
  01        # "/sensors/light"
  6e        # unsigned integer(value:9,"rt")
    2f73656e736f72732f6c    # text string(9 bytes)
    69676874          # "light-lux"
  09        # unsigned integer(value:10,"if")
  69        # "light-lux"
  6c        # text string(9 bytes)
    6c696768742d6c7578    # "light-lux"
  0a        # unsigned integer(value:10,"if")
```



```

66          # text string(6 bytes)
          73656e736f72          # "sensor"
a3          # map(# data item pairs:3)
01          # unsigned integer(value:1,"href")
78 23      # text string(35 bytes)
          687474703a2f2f777777
          2e6578616d706c652e63
          6f6d2f73656e736f7273
          2f74313233          # "http://www.example.com/sensors/t123"
03          # unsigned integer(value:3,"anchor")
6d          # text string(13 bytes)
          2f73656e736f72732f74
          656d70          # "/sensors/temp"
02          # unsigned integer(value:2,"rel")
6b          # text string(11 bytes)
          6465736372696265646279 # "describedby"
a3          # map(# data item pairs:3)
01          # unsigned integer(value:1,"href")
62          # text string(12 bytes)
          2f74          # "/t"
03          # unsigned integer(value:3,"anchor")
6d          # text string(13 bytes)
          2f73656e736f72732f74
          656d70          # "/sensors/temp"
02          # unsigned integer(value:2,"rel")
69          # text string(9 bytes)
          616c7465726e617465    # "alternate"

```

Figure 4: Web Links Encoded in CBOR

2.4.2. Link Format in JSON to CBOR Example

This examples shows conversion from link format JSON to CBOR format.

The JSON example from Section 2.4 becomes:

```

85          # array(number of data items:5)
a3          # map(# data item pairs:3)
01          # unsigned integer(value:1, "href")
68          # text string(8 bytes)
          2f73656e736f7273    # "/sensors"
0c          # unsigned integer(value:12,"ct")
18 28      # unsigned integer(value:40)
07          # unsigned integer(value:7,"title")
6c          # text string(12 bytes)
          53656e736f7220496e646578 # "Sensor Index"
a3          # map(# data item pairs:3)
01          # unsigned integer(value:1,"href")

```

```

6d      # text string(13 bytes)
        2f73656e736f72732f74
        656d70      # "/sensors/temp"
09      # unsigned integer(value:9,"rt")
6d      # text string(13 bytes)
        74656d70657261747572
        652d63      # "temperature-c"
0a      # unsigned integer(value:10,"if")
66      # text string(6 bytes)
        73656e736f72      # "sensor"
a3      # map(# data item pairs:3)
01      # unsigned integer(value:1,"href")
6e      # text string(14 bytes)
        2f73656e736f72732f6c
        69676874      # "/sensors/light"
09      # unsigned integer(value:9,"rt")
69      # text string(9 bytes)
        6c696768742d6c7578      # "light-lux"
0a      # unsigned integer(value:10,"if")
66      # text string(6 bytes)
        73656e736f72      # "sensor"
a3      # map(# data item pairs:3)
01      # unsigned integer(value:1,"href")
78 23   # text string(35 bytes)
        687474703a2f2f777777
        2e6578616d706c652e63
        6f6d2f73656e736f7273
        2f74313233      # "http://www.example.com/sensors/t123"
03      # unsigned integer(value:3,"anchor")
6d      # text string(13 bytes)
        2f73656e736f72732f74
        656d70      # "/sensors/temp"
02      # unsigned integer(value:2,"rel")
6b      # text string(11 bytes)
        6465736372696265646279      # "describedby"
a3      # map(# data item pairs:3)
01      # unsigned integer(value:1,"href")
62      # text string(12 bytes)
        2f74      # "/t"
03      # unsigned integer(value:3,"anchor")
6d      # text string(13 bytes)
        2f73656e736f72732f74
        656d70      # "/sensors/temp"
02      # unsigned integer(value:2,"rel")
69      # text string(9 bytes)
        616c7465726e617465      # "alternate"

```

Figure 5: Web Links Encoded in CBOR

3. Group Communication Management Objects in CBOR

3.1. Background

The CoAP Group Communications specification [RFC7390] defines group management objects in JSON format. These objects are used to represent IP multicast group information for CoAP endpoints. See [I-D.ietf-core-resource-directory] for more examples of using these objects.

3.2. Information Model

This section discusses the information model underlying the CoAP Group Communication management object payload.

A group membership JSON object contains one or more key/value pairs, and represents a single IP multicast group membership for the CoAP endpoint. Each key/value pair is encoded as a member of the JSON object, where the key is the member name and the value is the member's value.

The information model of the CoAP Group Communication management object can be summarized below:

```
collection = { * index => membership }
index = tstr .regexp "[A-Za-z0-9]{1,2}"
membership = {
  ? n: groupname,
  ? a: groupaddress,
}
groupname = tstr      ; host [ ":" port ]
groupaddress = tstr ; IPv4address [ ":" port ]
                  ; / "[" IPv6address "]" [ ":" port ]
```

Figure 6: CoAP Group Communication Data Model

3.3. Mapping

The objective of the mapping defined in this section is to map information from the JSON formats specified in [RFC7390] into CBOR format, using the rules of Section 4.2 of [RFC7049].

3.4. Group Communication Example

```
{ "8" :{ "a": "[ff15::4200:f7fe:ed37:14ca]" },
  "11":{ "n": "sensors.floor1.west.bldg6.example.com",
         "a": "[ff15::4200:f7fe:ed37:25cb]" },
  "12":{ "n": "All-Devices.floor1.west.bldg6.example.com",
         "a": "[ff15::4200:f7fe:ed37:abcd]:4567" }
}
```

Figure 7: Example from section 2.6.2.4 of [RFC7390]

becomes:

```

a3                                     # map(3)
  61                                 # text(1)
    38                             # "8"
  a1                                 # map(1)
    61                             # text(1)
      61                           # "a"
      78 1b                         # text(27)
        5b666631353a3a343230
        303a663766653a656433
        373a313463615d             # "[ff15::4200:f7fe:ed37:14ca]"
  62                                 # text(2)
    3131                           # "11"
  a2                                 # map(2)
    61                             # text(1)
      6e                           # "n"
      78 25                         # text(37)
        73656e736f72732e666c
        6f6f72312e776573742e
        626c6467362e6578616d
        706c652e636f6d             # "sensors.floor1.west.bldg6.example.com"
    61                             # text(1)
      61                           # "a"
      78 1b                         # text(27)
        5b666631353a3a343230
        303a663766653a656433
        373a323563625d             # "[ff15::4200:f7fe:ed37:25cb]"
  62                                 # text(2)
    3132                           # "12"
  a2                                 # map(2)
    61                             # text(1)
      6e                           # "n"
      78 29                         # text(41)
        416c6c2d446576696365
        732e666c6f6f72312e77
        6573742e626c6467362e
        6578616d706c652e636f
        6d                         # "All-Devices.floor1.west.bldg6.example.com"
    61                             # text(1)
      61                           # "a"
      78 20                         # text(32)
        5b666631353a3a343230
        303a663766653a656433
        373a616263645d3a34353637 # "[ff15::4200:f7fe:ed37:abcd]:4567"

```

Figure 8: Group Communication Management Object Encoded in CBOR

TO DO: Should the IP address/port number information be represented in a more compact way?

4. IANA Considerations

This specification registers the following additional Internet Media Types:

Type name: application

Subtype name: link-format+json

Required parameters: None

Optional parameters: None

Encoding considerations: Resources that use the "application/link-format+json" media type are required to conform to the "application/json" Media Type and are therefore subject to the same encoding considerations specified in [RFC7159], Section 11.

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): TEXT

Person & email address to contact for further information:
Carsten Bormann <cabo@tzi.org>

Intended usage: COMMON

Change controller: IESG

and

Type name: application

Subtype name: link-format+cbor

Required parameters: None

Optional parameters: None

Encoding considerations: Resources that use the "application/link-format+cbor" media type are required to conform to the "application/cbor" Media Type and are therefore subject to the same encoding considerations specified in [RFC7049], Section 7.

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): CBOR

Person & email address to contact for further information:
Kepeng Li <kepeng.lkp@alibaba-inc.com>

Intended usage: COMMON

Change controller: IESG

5. Security Considerations

The security considerations of [RFC6690], [RFC7049] and [RFC7159] apply.

(TBD.)

6. Acknowledgements

(TBD.)

Special thanks to Bert Greevenbosch who was an author on the initial version of a contributing document, as well as the original author on the CDDL notation.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, August 2012.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, October 2013.
- [RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

7.2. Informative References

- [I-D.greevenbosch-appsawg-cbor-cddl]
Vigano, C. and H. Birkholz, "CBOR data definition language: a notational convention to express CBOR data structures.", draft-greevenbosch-appsawg-cbor-cddl-06 (work in progress), July 2015.
- [I-D.ietf-core-resource-directory]
Shelby, Z., Koster, M., Bormann, C., and P. Stok, "CoRE Resource Directory", draft-ietf-core-resource-directory-03 (work in progress), June 2015.
- [I-D.pbryan-zyp-json-ref]
Bryan, P. and K. Zyp, "JSON Reference", draft-pbryan-zyp-json-ref-03 (work in progress), September 2012.
- [MNOT11] Nottingham, M., "Linking in JSON", November 2011,
<http://www.mnot.net/blog/2011/11/25/linking_in_json>.
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [RFC7390] Rahman, A. and E. Dijk, "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, October 2014.

Appendix A. Implementation

This appendix provides a simple reference implementation of the mapping between CoRE link format and Links-in-JSON.

(TBD - the reference implementation was used to create the above examples, but I still have to clean it up for readability and paste it in at 69 columns max.)

Authors' Addresses

Kepeng LI
Alibaba Group
Wenyixi Road, Yuhang District
Hangzhou, Zhejiang 311121
China

Email: kepeng.lkp@alibaba-inc.com

Akbar Rahman
InterDigital Communications, LLC
1000 Sherbrooke Street West
Montreal, Quebec H3A 3G4
Canada

Phone: +1-514-585-0761
Email: akbar.rahman@interdigital.com

Carsten Bormann (editor)
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

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