

CoRE
Internet-Draft
Intended status: Standards Track
Expires: June 13, 2011

Z. Shelby
Sensinode
December 10, 2010

CoRE Link Format
draft-ietf-core-link-format-02

Abstract

This document defines Web Linking using a link format for use by constrained web servers to describe hosted resources, their attributes and other relationships between links. Based on the HTTP Link Header format defined in [RFC5988](#), the CoRE Link Format is carried as a payload and is assigned an Internet media type. A well-known URI is defined as a default entry-point for requesting the links hosted by a server.

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 June 13, 2011.

Copyright Notice

Copyright (c) 2010 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

Internet-Draft

CoRE Link Format

December 2010

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Link Format	3
2.1.	Web Linking in CoRE	4
2.1.1.	Target and context URIs	4
2.1.2.	Link relations	5
2.1.3.	Use of anchors	5
2.2.	CoRE link extensions	5
2.2.1.	Resource name 'n' attribute	6
2.2.2.	Interface description 'd' attribute	6
2.2.3.	Content-type code 'ct' attribute	6
2.2.4.	Maximum size estimate 'sz' attribute	6
2.2.5.	Observable 'obs' attribute	7
3.	Well-known Interface	7
3.1.	Query Filtering	8
4.	Examples	9
5.	Security Considerations	10
6.	IANA Considerations	10
6.1.	Well-known 'core' URI	10
6.2.	New 'hosts' relation type	11
6.3.	New link-format Internet media type	11
7.	Acknowledgments	12
8.	Changelog	12
9.	References	14
9.1.	Normative References	14
9.2.	Informative References	14
	Author's Address	14

[1.](#) Introduction

The Constrained RESTful Environments (CoRE) working group aims at realizing the REST architecture in a suitable form for the most constrained nodes (e.g. 8-bit microcontrollers with limited RAM and ROM) and networks (e.g. 6LoWPAN). CoRE is aimed at machine-to-machine (M2M) applications such as smart energy and building automation.

The discovery of resources hosted by a constrained server is very important in machine-to-machine applications where there are no humans in the loop and static interfaces result in fragility. The discovery of resources provided by an HTTP Web Server is typically called Web Discovery and the description of relations between resources is called Web Linking [[RFC5988](#)]. In this document we refer to the discovery of resources hosted by a constrained web server, their attributes and other resource relations as CoRE Resource Discovery.

The main function of such a discovery mechanism is to provide URIs (called links) for the resources hosted by the server, complemented by attributes about those resources and possible further link relations. In CoRE this collection of links is carried as a resource of its own (as opposed to as HTTP headers delivered with a specific resource). This document specifies a link format for use in CoRE Resource Discovery by extending the HTTP Link Header Format [[RFC5988](#)] to describe these link descriptions. The CoRE Link Format is carried as a payload and is assigned an Internet media type. A well-known URI `"/.well-known/core"` is defined as a default entry-point for requesting the list of links about resources hosted by a server, and thus performing CoRE Resource Discovery.

[2.](#) Link Format

The CoRE Link Format extends the HTTP Link Header format specified in

[RFC5988], which is specified in Augmented Backus-Naur Form (ABNF) notation. The format does not require special XML or binary parsing, is fairly compact, and is extensible – all important characteristics for CoRE. It should be noted that this link format is just one serialization of typed links defined in [RFC5988], others include Atom or HTTP links. It is expected that resources discovered in the CoRE Link Format may also be made available in alternative formats on the greater Internet.

[Section 5 of \[RFC5988\]](#) did not require an Internet media type for the defined link format, as it was defined to be carried in an HTTP header. This specification thus defines a Internet media type for

the CoRE Link Format (see [Section 6.3](#)).

The CoRE link format uses the ABNF description and associated rules in [Section 5 of \[RFC5988\]](#). In addition, the pchar rule is taken from [\[RFC3986\]](#). The "Link:" text is omitted as that is part of the HTTP Link Header. As in [\[RFC5988\]](#), multiple link descriptions are separated by commas. Note that commas can also occur in quoted strings and URIs but do not end a description. The CoRE link format MUST use UTF-8 encoding, which SHOULD be in NFC (Unicode Normalization Form C). See [Section 3 of \[RFC5198\]](#), which explains why it useful to represent Unicode in a single unique form.

[2.1](#). Web Linking in CoRE

This link format is used for a similar purpose to that described in [\[RFC5988\]](#), to describe relationships between resources via typed links, so-called "Web Linking". However in this specification Web Linking is extended with specific constrained M2M attributes, links are carried as a message payload rather than in an HTTP Link Header, and a default interface is defined to discover resources hosted by a server. This specification defines a new relation type "hosts", which indicates that the resource is hosted by the server from which the link document was requested.

[2.1.1](#). Target and context URIs

Each link conveys one target URI as a URI-reference inside angle brackets ("[<>](#)"). The context URI of a link (also called base URI in [\[RFC3986\]](#)) conveyed in the CoRE Link Format is by default built from

the scheme and authority parts of the target URI. In the absence of this information in the target URI, the context URI is built from the scheme and authority that was used for referencing the resource returning the set of links, replacing the path with an empty path. Thus by default links can be thought of as describing a target resource hosted by the server. Other relations can be expressed by including an anchor parameter (which defines the context URI) along with an explicit relation parameter. This is an important difference to the way the HTTP Link Header format is used, as it is included in the header of an HTTP response for some URI (this URI is by default the context URI). Thus the HTTP Link Header is by default relating the target URI to the URI that was requested. In comparison, the CoRE link format includes one or more links, each describing a resource hosted by a server by default. Other relations can be expressed by using the anchor parameter. See [Section 5 of \[RFC3986\]](#) for a description of how URIs are constructed from URI references.

[2.1.2.](#) Link relations

Since links in the CoRE Link Format are typically used to describe resources hosted by a server, and thus in the absence of the relation parameter the new relation type "hosts" is assumed (see [Section 6.2](#)). The "hosts" relation type indicates that the target URI is a resource hosted by the server given by the base URI, or, if present, the anchor parameter.

To express other relations a links can make use of any registered relation parameter or target attributes by including the relation parameter. The context of a relation can be defined using the anchor parameter. In this way, relations between resources hosted on a server, or between hosted resources and external resources can be expressed.

[2.1.3.](#) Use of anchors

As per [Section 5.2 of \[RFC5988\]](#) a link description MAY include an "anchor" attribute, in which case the context is the URI included in that attribute. This is used to describe a relationship between two resources. A consuming implementation can however choose to ignore

such links. It is not expected that all implementations will be able to derive useful information from explicitly anchored links.

[2.2.](#) CoRE link extensions

The following CoRE specific target attributes are defined. These attributes describe information useful in accessing the target link of the relation, and in some cases may be URIs. These URIs MUST be treated as indicators, and are not meant to be actually retrieved like a URL. When attributes are compared, they MUST be compared as strings. Relationships to resources that are meant to be retrieved should be expressed as separate links using the anchor attribute and the appropriate relation type.

```
link-extension    = ( "n" "=" quoted-string )
link-extension    = ( "d" "=" <"> URI-reference <">)
link-extension    = ( "ct" "=" integer )
link-extension    = ( "sz" "=" integer )
link-extension    = ( "obs" )
integer           = 1*DIGIT
```

[2.2.1.](#) Resource name 'n' attribute

The resource name "n" attribute is used to assign a semantically important name to a resource. In the case of a temperature sensor resource this could be an application-specific semantic name like "TemperatureC", a URN like "urn:temperature:centigrade" or a URI referencing a specific concept in an ontology like "http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature". Multiple name attributes MAY appear in a link.

The name attribute is not meant to be used to assign a human readable name to a resource. The "title" attribute defined in [[RFC5988](#)] is meant for that purpose.

[2.2.2.](#) Interface description 'd' attribute

The interface description "d" attribute is used to provide a URI or URN indicating a specific interface definition used to interact with the target resource. This could be for example the URI of a WADL definition of the target resource "http://www.example.org/sensor.wadl", or a URN indicating the type of interface to the resource "urn:sensor". Multiple description attributes MAY appear in a link.

[2.2.3.](#) Content-type code 'ct' attribute

The Content-type code "ct" attribute provides a hint about the Internet media type this resource returns. Note that this is only a hint, and does not override the Content-type Option of a CoAP response obtained by actually following the link. The value is in the CoAP identifier code format as a decimal ASCII integer [[I-D.ietf-core-coap](#)]. For example application/xml would be indicated as "ct=41". If no Content-type code attribute is present then nothing about the type can be assumed. The Content-type code attribute MUST NOT appear more than once in a link.

Alternatively, the "type" attribute MAY be used to indicate an Internet media type as a quoted-string [[RFC5988](#)]. It is not however expected that constrained implementations are able to parse quoted-string Content-type values. A link MAY include either a ct attribute or a type attribute, but MUST NOT include both.

[2.2.4.](#) Maximum size estimate 'sz' attribute

The maximum size estimate attribute "sz" gives an indication of the maximum size of the link indicated by the target URI. This attribute is not expected to be included for small resources that can comfortably be carried in a single MTU, but SHOULD be included for

resources larger than that. The maximum size estimate attribute MUST NOT appear more than once in a link.

[2.2.5.](#) Observable 'obs' attribute

The observable attribute "obs" is used to indicate that this resource supports the observation feature defined in [[I-D.ietf-core-observe](#)]. This attribute is used as a flag, and thus it has no value component.

The observable attribute MUST NOT appear more than once in a link.

3. Well-known Interface

Resource discovery in CoRE is accomplished through the use of a well-known resource URI which returns a list of links about resources hosted by that server and other link relations. Well-known resources have a path component that begins with `"/.well-known/"` as specified in [\[RFC5785\]](#). This document defines a new well-known resource for CoRE Resource Discovery `"/.well-known/core"`. A server implementing this specification MUST support this resource on the default port appropriate for the protocol for the purpose of resource discovery. It is however up to the application which links are included and how they are organized. In the absence of any links, a zero-length payload is returned. The resource representation of this resource uses the CoRE Link Format described in [Section 2](#). URIs under the path `/.well-known/core` are reserved by this specification for use by servers in organizing links into multiple resources, each with their own URI.

The CoRE resource discovery interface supports the following interactions:

- o Performing a GET on `/.well-known/core` to the default port returns a set of links available from the CoAP server (if any) in the CoRE Link Format. These links might describe resources hosted on that server, on other servers, or express other kinds of link relations as described in [Section 2](#).
- o Filtering may be performed on any of the link format attributes using a query string as specified in [Section 3.1](#). For example `[GET /.well-known/core?n=TemperatureC]` would request resources with the name `TemperatureC`. A server is not however required to support filtering.
- o More capable servers such as proxies could support a resource directory by requesting the resource descriptions of other endpoints or allowing servers to POST requests to `/.well-known/core`. The details of such resource directory functionality is however

separately.

End-points with a large number of resources SHOULD include resource descriptions only for important links and MAY organize their resource descriptions into a hierarchy of link resources. This is done by including links in the /.well-known/core list which point to other resource lists using the "section" relation type, e.g. </.well-known/core/sensors>. Such a hierarchy MUST be under the /.well-known/core path.

[3.1.](#) Query Filtering

A server implementing this document MAY recognize the query part of a resource-discovery URI as a filter on the resources to be returned. The query part should conform to the following syntax. Note that this only defines querying for a single parameter at a time.

```
filter-query = resource-param "=" query-pattern
resource-param = "uri" | parmname
query-pattern = 1*pchar [ "*" ]
```

The resource-param "uri" refers to the URI-reference between the "<" and ">" characters of a link. Other resource-param values refer to the link attribute they name. Filtering is performed by comparing the query-pattern against the value of the attribute identified by the resource-param for each link-value in the collection of resources identified by the URI path.

If the decoded query-pattern does not end with "*", a link value matches the query only if the value of the attribute or URI-reference denoted by the resource-param is bitwise identical to the query-pattern. If the decoded query-pattern ends with "*", it is sufficient that the remainder of the query-pattern be a prefix of the value denoted by the resource-param.

It is not expected that very constrained nodes support filtering. Implementations not supporting filtering MUST simply ignore the query string and return the whole resource for unicast requests. An implementation supporting filtering MUST return only those entries for which the value of the link attribute identified by the resource-param contains the query-pattern as a substring. If resource descriptions are organized hierarchically, a query on the root resource /.well-known/core MUST return all matching resource descriptions from the entire hierarchy.

When using a transfer protocol like CoAP that supports multicast requests, special care is taken. A multicast request with a query string **MUST** not be responded to if filtering is not supported (to avoid a needless response storm).

[4.](#) Examples

A few examples of typical link descriptions in this format follows. Multiple resource descriptions in a representation are separated by commas. Linefeeds never occur in the actual format, but are shown in the example for readability.

This example includes links to two different sensors.

```
REQ: GET /.well-known/core
```

```
RES: 200 OK
</sensors/temp>;ct=41;n="TemperatureC",
</sensors/light>;ct=41;n="LightLux"
```

This example arranges link descriptions hierarchically, with the entry point including a link to a sub-resource containing links about the sensors.

```
REQ: GET /.well-known/core
```

```
RES: 200 OK
</.well-known/core/sensors>;rel="section";ct=40
```

```
REQ: GET /.well-known/core/sensors
```

```
RES: 200 OK
</sensors/temp>;n="TemperatureC",
</sensors/light>;ct=41;n="LightLux"
```

An example query filter may look like:

```
REQ: GET /.well-known/core?n=LightLux
```

```
RES: 200 OK
</sensors/light>;ct=41;n="LightLux"
```

This example shows the use of an anchor attribute to relate the temperature sensor resource to an external description and to an

alternative URL.

REQ: GET /.well-known/core

RES: 200 OK

```
</sensors>;rel="index";n="Sensor Index",  
</sensors/temp>;n="TemperatureC",  
</sensors/light>;ct=41;n="LightLux",  
<http://www.example.com/sensors/temp123>;anchor="/sensors/temp"  
;rel=describedby,  
</t>;anchor="/sensors/temp";rel=alternate
```

If a client is interested to find relations about a particular resource, it can perform a query on the anchor parameter:

REQ: GET /.well-known/core?anchor=/sensors/temp

RES: 200 OK

```
<http://www.example.com/sensors/temp123>;anchor="/sensors/temp"  
;rel=describedby,  
</t>;anchor="/sensors/temp";rel=alternate
```

[5.](#) Security Considerations

This document needs the same security considerations as described in [Section 7 of \[RFC5988\]](#). The /.well-known/core resource may be protected e.g. using DTLS when hosted on a CoAP server as per [\[I-D.ietf-core-coap\] Section 10.2.](#)

Multicast requests using CoAP for the well-known link-format resources could be used to perform denial of service on a constrained network. A multicast request SHOULD only be accepted if the request is sufficiently authenticated and secured.

CoRE link format parsers should be aware that a link description may be cyclical, i.e., contain a link to itself. These cyclical links could be direct or indirect (i.e., through referenced link resources). Care should be taken when parsing link descriptions and

accessing cyclical links.

[6.](#) IANA Considerations

[6.1.](#) Well-known 'core' URI

This memo registers the "core" well-known URI in the Well-Known URI Registry as defined by [\[RFC5785\]](#).

Shelby

Expires June 13, 2011

[Page 10]

Internet-Draft

CoRE Link Format

December 2010

URI suffix: core

Change controller: IETF

Specification document(s): [\[\[this document \]\]](#)

Related information: None

[6.2.](#) New 'hosts' relation type

This memo registers the new "hosts" Web Linking relation type as per [\[RFC5988\]](#).

Relation Name: hosts

Description: Refers to a resource hosted by the server indicated by the link context.

Reference: [\[\[this document \]\]](#)

Notes: This relation is used in CoRE where links are retrieved as a `/.well-known/core` resource representation, and by default the context of the links is the server at `coap://authority` from which `/.well-known/core` was requested.

Application Data: None

[6.3.](#) New link-format Internet media type

This memo registers the a new Internet media type for the CoRE link format, `application/link-format`.

Type name: application

Subtype name: link-format

Required parameters: None

Optional parameters: The query string may contain uri= to match the URI, or any other attribute defined for the link format to match that attribute.

Encoding considerations: UTF-8 (NFC)

Security considerations: None

Interoperability considerations:

Shelby

Expires June 13, 2011

[Page 11]

Internet-Draft

CoRE Link Format

December 2010

Published specification: [[this document]]

Applications that use this media type: CoAP server and client implementations for resource discovery and HTTP applications that use the link-format as a payload.

Additional information:

Magic number(s):

File extension(s):

Macintosh file type code(s):

Intended usage: COMMON

Restrictions on usage: None

Author: CoRE WG

Change controller: IETF

[7.](#) Acknowledgments

Special thanks to Peter Bigot, who has made a considerable number reviews and text contributions that greatly improved the document. In particular, Peter is responsible for the ABNF descriptions and the idea for a new "hosts" relation type.

Thanks to Mark Nottingham and Eran Hammer-Lahav for discussions and ideas that led to this draft, and to Carsten Bormann and Martin Thomson for extensive comments and contributions that improved the text.

Thanks to Michael Stuber, Richard Kelsey, Cullen Jennings, Guido Moritz, Peter Van Der Stok, Adriano Pezzuto, Lisa Dussealt, Alexey Melnikov, Gilbert Clark, Salvatore Loreto, Petri Mutka, Szymon Sasin, Robert Quattlebaum, Robert Cragie, Angelo Castellani, Tom Herbst, Ed Berozet, Gilman Tolle, Robby Simpson, Colin O'Flynn and David Ryan for helpful comments and discussions that have shaped the document.

8. Changelog

Changes from ietf-01 to ietf-02:

Shelby

Expires June 13, 2011

[Page 12]

Internet-Draft

CoRE Link Format

December 2010

- o Added references to [RFC5988](#) (#41).
- o Removed sh and id link-extensions (#42).
- o Defined the use of UTF-8 (#84).
- o Changed query filter definition for any parameter (#70).
- o Added more example, now as a separate section (#43).
- o Mentioned cyclical links in the security section (#57).
- o Removed the sh and id attributes, added obs and sz attributes (#42).
- o Improved the context and relation description wrt [RFC5988](#) and requested a new "hosts" default relation type (#85).

Changes from ietf-00 to ietf-01:

- o Editorial changes to correct references.
- o Formal definition for filter query string.
- o Removed URI-reference option from "n" and "id".
- o Added security text about multicast requests.

Changes from shelby-00 to ietf-00:

- o Fixed the ABNF link-extension definitions (quotes around URIs, integer definition).
- o Clarified that filtering is optional, and the query string is to be ignored if not supported (and the URL path processed as normally).
- o Required support of wildcard * processing if filtering is supported.
- o Removed the assumption of a default content-type assumption.

[9.](#) References

Shelby	Expires June 13, 2011	[Page 13]
--------	-----------------------	-----------

Internet-Draft	CoRE Link Format	December 2010
----------------	------------------	---------------

[9.1.](#) Normative References

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC5988] Nottingham, M., "Web Linking", [RFC 5988](#), October 2010.

[9.2.](#) Informative References

[I-D.ietf-core-coap]

Shelby, Z., Frank, B., and D. Sturek, "Constrained Application Protocol (CoAP)", [draft-ietf-core-coap-03](#) (work in progress), October 2010.

[I-D.ietf-core-observe]

Hartke, K. and Z. Shelby, "Observing Resources in CoAP", [draft-ietf-core-observe-00](#) (work in progress), October 2010.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", [RFC 5198](#), March 2008.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), April 2010.

Author's Address

Zach Shelby
Sensinode
Kidekuja 2
Vuokatti 88600
FINLAND

Phone: +358407796297
Email: zach@sensinode.com