CoRE Link Format
draft-shelby-core-link-format-00

Abstract

This document defines a link format for use by constrained CoAP web servers to describe URIs of resources offered along with other attributes. Based on the HTTP Link Header format, 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 list of links to resources hosted by a server.

Status of this Memo

Copyright Notice

Table of Contents

---

1. Introduction [TOC](#)

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
[I-D.shelby-core-coap-req] (Shelby, Z., Stuber, M., Sturek, D., Frank,
B., and R. Kelsey, "CoAP Requirements and Features," April 2010.).
The discovery of resources offered 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. In this document we refer to the discovery of resources
offered by a CoAP server as resource discovery.
The core function of such a discovery mechanism is to provide URIs
("links") for the resources offered, complemented by information
describing the relationship between the resource description and each
resource as well as other attributes. When such a collection of
attributed resource references (links) is offered as a resource of its

own (as opposed to as HTTP headers delivered with a different resource), we speak of its representation as a link-format. This document specifies a link-format for use in CoRE resource discovery by extending the HTTP Link Header Format [I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.) to describe resources hosted by a constrained server. 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 to resources hosted by a server.

---

2. Link Format

CoRE resource discovery extends the HTTP Link Header format specified in [I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.) which is specified in Augmented Backus-Naur Form (ABNF) notation [RFC5234] (Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," January 2008.). The format does not require special XML or binary parsing, and is extensible.
This link format is used for a similar purpose to that described in [I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.), to describe one or more relationships between resources. However in this specification the link format is extended with specific constrained M2M link parameters, links are carried as a payload rather than in a message header, and a default interface is defined to discover resources described by these links.
[I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.) did not require an Internet media type for this link format, as it assumes to be carried in an HTTP header. This specification thus requests a Internet media type for this format (see Section 5.2 (New link-format Internet media type)).
The CoRE link format uses the ABNF description and associated rules in Section 5 of [I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.). The "Link:" text is omitted as that is part of the HTTP Link Header. Multiple link descriptions are separated by commas. The CoRE link format MUST use the US-ASCII character set (support for RFC2231 encoding of non-ASCII content TBD). The following CoRE specific link-extension parameters to the format are defined:

```
        link-extension    = ( "d" "=" URI )
        link-extension    = ( "sh" "=" URI-Reference )
        link-extension    = ( "n" "=" ( quoted-string | URI ) )
        link-extension    = ( "ct" "=" integer )
        link-extension    = ( "id" "=" ( quoted-string | URI ) )
```

2.1. Target and context IRIs

Each link description conveys one target URI as a URI-Reference inside angle brackets ("<>"). The context of a link conveyed in the description is by default the URI of the resource that returned the link-format representation (usually ./well-known/core). Thus each link can be thought of as describing a target resource hosted by the server in the absence of further relation information. 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). 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 link entries, each describing a resource hosted by a server.
As per Section 5.2 of [I-D.nottingham-http-link-header] (Nottingham, M., "Web Linking," May 2010.) a link description MAY include an "anchor" attribute, in which case the context is the URI included in that attribute. This can be used to describe a relationship between two resources. A consuming implementation can however choose to ignore such links. It is not expected that most implementations will be able to derive useful from explicitly anchored links.

2.2. Link relation 'rel' usage

Link descriptions in CoRE are typically used to describe entry points to services hosted by the server, and thus in the absence of the rel attribute the registered "service" relation type is assumed. In the CoRE link format the service relation type indicates that the link is a service hosted by the server (in the absence of the anchor attribute). A description can make use of any registered relation type or extension types in the form of a URI by including the rel attribute.

2.3. Description 'd' usage

The description "d" attribute can provide a URI to a specific interface definition used to access the target resource. This could be for example a URI to the WADL definition of the target resource. Multiple description attributes MAY appear in a link description.

2.4. Alternative URI 'sh' usage

This attribute can be included to define an alternative short URI which can also be used to access the target resource. Multiple alternative short URI attributes MAY appear in a link description.

2.5. Resource name 'n' usage

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

2.6. Content-type code 'ct' usage

The Content-type code "ct" attribute provides a hint about the Internet media type this resource returns. The value is in the CoAP identifier code format as a decimal ASCII integer [I-D.ietf-core-coap] (Shelby, Z., Frank, B., and D. Sturek, "Constrained Application Protocol (CoAP)," September 2010.). For example application/xml would be indicated as "ct=41". If no Content-type code attribute is present then text/plain is assumed. The Content-type code attribute MUST NOT appear more than once in a link description.
Alternatively, the "type" attribute MAY be used to indicate an Internet media type as a quoted-string. It is not however expected that constrained implementations are able to parse quoted-string Content-type values.

2.7. Resource identifier 'id' usage

The resource identifier "id" field is a unique identifier (e.g. UUID) for this resource for use in e.g. resource or search directories. This attribute may be in quoted-string format (e.g. in the case of a UUID or XRI) or in URI format (e.g. in the case of a URN). The resource identifier attribute MUST NOT appear more than once in a link description.

2.8. Examples

A few examples of typical link descriptions in this format follows. Multiple resource descriptions in a representation are separated by commas. Commas can also occur in quoted strings and URIs but do not end a description. Linefeeds never occur in the actual format, but are shown in the example for readability.
This example includes link descriptions for an index to sensors hosted by a server, along with links two two different sensors.

```
GET /.well-known/core

</sensors>;rel="index";n="Sensor Index",
</sensors/temp>;sh="/t";n="TemperatureC",
</sensors/light>;sh="/l";ct=41;n="LightLux"
```

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

```
GET /.well-known/core

</.well-known/core/sensors>;rel="section"
;type="application/link-format"

GET /.well-known/core/sensors

</sensors/temp>;sh="/t";n="TemperatureC",
</sensors/light>;sh="/l";ct=41;n="LightLux"
```

---

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 (resource descriptions) offered by that constrained server. Well-known resources have a reserved base URI "/.well-known/" as specified in [RFC5785] (Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)," April 2010.). This document defines a new well-known URI for CoRE discovery "/.well-known/core" Section 5.1 (Well-known 'core' URI). A server implementing this specification MUST support this URI on the default port appropriate for the protocol, for the purpose of resource discovery. It is however up to the application which link descriptions are included and how they are organized. In the absense of any links, a zero-length payload is returned. The resource

representation of this resource is described in [Section 2 (Link Format)](#).

The CoRE resource discovery interface supports the following interactions:

*Performing a GET on /.well-known/core to the default port returns a list of link descriptions available from a CoAP server (if any).

*Filtering may be performed on any of the link format attributes using a query string as specified in [Section 3.1 (Query Filtering)](#). For example [GET /.well-known/core?n=TemperatureC] would request resources with the name TemperatureC. A server is not however required to support filtering.

*More capable servers such as proxies could support a resource directory by requesting the resource descriptions of other end-points or accepting [POST /.well-known/core messages] from other servers. This adds the resources of other end-points as a sub-resource in which absolute URIs are included for the link-values. The details of such resource directory functionality is however out of scope for this document.

End-points with a large number of resources SHOULD include resource descriptions only for important services or collections and 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, e.g. </.well-known/core/sensors>. Such a hierarchy SHOULD be under the /.well-known/core path but could be located elsewhere.

3.1. Query Filtering

A server implementing this document MAY support the query string /.well-known/core? with uri= corresponding to the link-value or any of the resource description attributes for the purpose of filtering a discovery. It is not expected that simple implementations support filtering, but instead will just ignore the query string. Wildcard * endings MAY be supported. An exact match is performed on the query string, and a 200 OK response is returned with link descriptions that contains the matching entries (if any). If resource descriptions are organized hierarchically, a query on the root resource /.well-known/core SHOULD return all matching resource descriptions from the entire hierarchy. An example query on the example link descriptions from [Section 2 (Link Format)](#) may look like:

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

     </sensors/light>;sh="/l";ct=41;n="LightLux"
```

---

4. Security Considerations

This document needs the same security considerations as described in
Section 7 of [I-D.nottingham-http-link-header] (Nottingham, M., "Web
Linking," May 2010.). The /.well-known/core resource may be protected
e.g. using DTLS when hosted on a CoAP server.

---

5. IANA Considerations

5.1. Well-known 'core' URI

This memo registers the "core" well-known URI in the Well-Known URI
Registry as defined by [RFC5785].
URI suffix: core
Change controller: IETF
Specification document(s): [[this document]]
Related information: None

5.2. 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: US-ASCII
Security considerations: None
Interoperability considerations:
Published specification: [[this document]]

Applications that use this media type: CoAP server and client
implementations.
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

---

6. Acknowledgments

Special thanks to Mark Nottingham and Eran Hammer-Lahav for discussions
and ideas that led to this draft, and to Carsten Bormann 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
Beroset, Gilman Tolle, Robby Simpson, Peter Bigot, Colin O'Flynn and
David Ryan for helpful comments and discussions that have shaped the
document.

---

7. Changelog

---

8. References

8.1. Normative References

[I-D.ietf-core-coap]	Shelby, Z., Frank, B., and D. Sturek, "Constrained Application Protocol (CoAP)," draft-ietf-core-coap-02 (work in progress), September 2010 (TXT).
[I-D.nottingham-http-link-header]	Nottingham, M., "Web Linking," draft-nottingham-http-link-header-10 (work in progress), May 2010 (TXT).

[RFC5234]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," STD 68, RFC 5234, January 2008 (TXT).
[RFC5785]	Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)," RFC 5785, April 2010 (TXT).

8.2. Informative References

[I-D.shelby-core-coap-req]	Shelby, Z., Stuber, M., Sturek, D., Frank, B., and R. Kelsey, "CoAP Requirements and Features," draft-shelby-core-coap-req-01 (work in progress), April 2010 (TXT).

Author's Address

	Zach Shelby
	Sensinode
	Kidekuja 2
	Vuokatti 88600
	FINLAND
Phone:	+358407796297
Email:	zach@sensinode.com