Abstract

The "Thing-to-Thing Data Hub" is a RESTful, hypermedia-driven Web application that can be used in Thing-to-Thing communications to share data items such as thing descriptions, configurations, resource descriptions, or firmware updates at a central location.

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 https://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 9, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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.
1. Introduction

In Thing-to-Thing communication, there is often a need to share data items of common interest through a central location. For example, the Resource Directory [I-D.ietf-core-resource-directory] aggregates descriptions of Web resources held on constrained nodes, which enables other nodes to easily discover these resources; a Thing Directory [W3C.CR-wot-architecture-20190516] stores metadata of IoT devices, allowing clients to discover interaction affordances and supported protocol bindings of Things; and a Firmware Server [I-D.ietf-suit-architecture] stores firmware images and manifests, making this data available to deployed devices, commissioning tools, and other services.

As more and more Thing-to-Thing applications are implemented, it becomes increasingly important being able to not only share resource descriptions and firmware updates but also many other kinds of data, such as default configurations for new devices, service locations, or certificate revocation lists. Resource directories and firmware servers are not a good fit for these kinds of data, as they’re specialized to their use cases and generally not accepting any other kinds of data. And creating new, specialized applications for every type of data is not practical in the long term.

This document defines a simple "data hub" application, a RESTful Web application with a machine-understandable hypermedia API. A "data hub" generalizes the concept of a central repository for different applications and is suitable for constrained environments [RFC7228]. Specifically, it enables clients to share data items in any format and provides means for creating, reading, observing, updating, deleting, and finding data items at a data hub server.

Data hubs are intended to be used primarily with Constrained Application Protocol (CoAP) [RFC7252].
Features:

- General
  
The data hub generalizes the concept of a directory or repository to data items of any Internet media type. This means that applications using the data hub aren’t stuck forever with the same media types or limited to just resource descriptions or firmware updates.

- Searchable
  
  Clients can retrieve a subset of data items from a data hub based on item metadata.

- Observable
  
  Data items published to a data hub are exposed as resources. As such, they can be observed for changes [RFC7641] over CoAP. This allows clients to stay informed of information that other clients update over time. As a result, the data hub functions similar to a Publish-Subscribe Broker [I-D.ietf-core-coap-pubsub].

- Evolvable
  
  The key differentiator of the data hub compared to Resource Directory [I-D.ietf-core-resource-directory] and CoAP Publish-Subscribe Broker [I-D.ietf-core-coap-pubsub] lies in the evolvability of the application -- the ability to respond effectively to the need for changes without negatively impacting existing and new clients.

  Data hubs enable fine-grained evolvability by driving all interactions by machine-understandable hypermedia elements. Features can be added, changed or removed in a safe, backwards-compatible way simply by updating the data hub representation to expose appropriate links and forms.

1.1. Notational Conventions

  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].

  Terms defined in this document appear in _cursive_ where they are introduced.
2. Data Model

The data model of the "Thing-to-Thing Data Hub" application consists of three elements: the _data hub_ itself, a _data collection_, and a number of _data items_ that have been shared (Figure 1).

![Figure 1: A Data Collection with a Number of Shared Data Items Hosted at a Data Hub]

Data Hub

A data hub is a Web application running on a Web server that hosts the data collection and the data items.

Data Collection

A data collection is a collection resource that contains the data items.

Representations of data collections MUST have the "application/coral+cbor" media type [I-D.hartke-t2trg-coral]. The representations consist primarily of links to the data items. These links have the `<http://www.iana.org/assignments/relation/item>` link relation type [RFC6573]. To reduce the number of round-trips, the representations MAY also embed (complete or partial) representations of the data items. Forms contained in the representation enable interactions with the data collection and the data items, as detailed in the following sections. The representations MAY additionally contain other links and forms that are not described in this document, such as a link with the `<http://www.iana.org/assignments/relation/alternate>` link relation type [W3C.REC-html52-20171214] that references an alternate representation of the data collection resource.
In this version of this document, a data hub is defined to have a depth of only one level. That is, all data item resources are organized directly under the top-level data collection resource. This could be extended to multiple levels in a future version.

Data Item

A data item is a resource that is a member of the data collection resource.

Data item representations can have any media type. However, a data collection MAY restrict the media types it accepts for publication. In this case, the form in the representation of the data collection for creating data items MUST list the acceptable media types using form fields of type <http://coreapps.org/coap#accept>.

The representations of data items MAY link back to the data collection resource using the <http://www.iana.org/assignments/relation/collection> link relation type [RFC6573].

3. Interaction Model

The interaction model consists of eight potential interactions with a data hub: discovering and reading the data collection, and creating, reading, observing, updating, deleting, and finding shared data items in the data collection.

Discovering a Data Hub

In this version of this document, clients are assumed to be pre-configured with an entry-point IRI for a data collection at a data hub.

Reading a Collection

A client can retrieve a representation of a data collection by dereferencing the entry-point IRI. As described above, the representation of the data collection includes links to (and, optionally, representations of) the data items in the data collection. The representation of the data collection also includes forms for creating, updating, deleting, and finding data items.

Creating an Item
The representation of a data collection MAY contain a form with the `<http://coreapps.org/collections#create>` operation type. Submitting this form with a representation in one of the acceptable media types creates a new data item in the data collection. The acceptable media types are indicated by form fields of type `<http://coreapps.org/coap#accept>`.

Implementations of this version of this document MUST use the method implied by the `<http://coreapps.org/collections#create>` operation type, i.e., the POST method [RFC7252]. A form indicating different a method MUST be ignored.

On success, the location of the created data item MUST be conveyed in a 2.01 (Created) response using the Location-Path and Location-Query options [RFC7252].

Reading an Item

A client can retrieve a representation of a data item by following a link with the `<http://www.iana.org/assignments/relation/item>` link relation type in the representation of the data collection.

Observing an Item

A client can observe a data item by following a link with the `<http://www.iana.org/assignments/relation/item>` link relation type in the representation of the data collection and observing the target resource as specified in RFC 7641 [RFC7641].

Updating an Item

For each data item in a data collection, the representation of the data collection MAY include a form with the `<http://coreapps.org/base#update>` operation type nested within the link to the data item. Submitting this form updates the data item to the submitted representation.

Implementations of this version of this document MUST use the method implied by the `<http://coreapps.org/base#update>` operation type, i.e., the PUT method [RFC7252]. A form indicating different a method MUST be ignored.

On success, a 2.04 (Changed) response is returned.

Deleting an Item

For each data item in a data collection, the representation of the data collection MAY include a form with the `<http://coreapps.org/`
The data hub application does not define any specific mechanisms for protecting the confidentiality and integrity of messages exchanged between a data hub and a client. It is recommended that implementations employ application layer or transport layer mechanisms for interactions with a data hub.

The data hub application does not define any specific mechanisms for protecting the confidentiality and integrity of representations of data items shared through a data hub. For scenarios where end-to-end...
security matters, such as for firmware updates
[I-D.ietf-suit-information-model], implementations should employ an
object security mechanism.

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[I-D.hartke-t2trg-coral]
Hartke, K., "The Constrained RESTful Application Language (CoRAL)", draft-hartke-t2trg-coral-09 (work in progress),
July 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,


[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252,

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641,

[RFC8132] van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and FETCH Methods for the Constrained Application Protocol (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017,

6.2. Informative References

[I-D.ietf-core-coap-pubsub]
Acknowledgements

Thanks to Christian Amsuess and Jaime Jimenez for helpful comments and discussions that have shaped the document.

Author’s Address
Klaus Hartke  
Ericsson  
Torshamnsgatan 23  
Stockholm SE-16483  
Sweden  

Email: klaus.hartke@ericsson.com