

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 2, 2014

I. Young, Ed.
Independent
December 29, 2013

Metadata Query Protocol
draft-young-md-query-01

Abstract

This document defines a simple protocol for retrieving metadata about named entities, or named collections of entities. The goal of the protocol is to profile various aspects of HTTP to allow requesters to rely on certain, rigorously defined, behaviour.

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 July 2, 2014.

Copyright Notice

Copyright (c) 2013 IETFTrust 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.

Table of Contents

1.	Introduction	2
1.1.	Notation and Conventions	3
1.2.	Terminology	3
2.	Protocol Transport	3
2.1.	Transport Protocol	3
2.2.	HTTP Version	3
2.3.	HTTP Method	3
2.4.	Request Headers	3
2.5.	Response Headers	4
2.6.	Status Codes	4
2.7.	Base URL	5
2.8.	Content Negotiation	5
3.	Metadata Query Protocol	5
3.1.	Identifiers	6
3.2.	Protocol	7
3.2.1.	Request	7
3.2.2.	Response	7
3.2.3.	Example Request and Response	7
4.	Efficient Retrieval and Caching	8
4.1.	Conditional Retrieval	8
4.2.	Content Caching	8
4.3.	Content Compression	8
5.	Protocol Extension Points	8
6.	Security Considerations	9
6.1.	Integrity	9
6.2.	Confidentiality	9
6.3.	Authentication	9
7.	IANA Considerations	9
8.	Acknowledgements	9
9.	Normative References	10
Appendix A.	Change Log (to be removed by RFC Editor before publication)	10
A.1.	Since draft-young-md-query-00	10
A.2.	Since draft-lajoie-md-query-01	11

[1.](#) Introduction

Many clients of web-based services are capable of consuming descriptive metadata about a service in order to customize or information the client's connection parameters. While the form of the metadata (e.g., JSON, XML) and content varies between services

this document specifies a set of semantics for HTTP [[RFC2616](#)] that allow clients to rely on certain behavior. The defined behavior is meant to make it easy for clients to perform queries, to be efficient for both requesters and responders, and to allow the responder to scale in various ways.

[1.1.](#) Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[BCP14](#)].

This document makes use of the Augmented BNF metalanguage defined in [[STD68](#)].

[1.2.](#) Terminology

entity - A single logical construct for which metadata may be asserted. Generally this is a network accessible service.

metadata - A machine readable description of certain entity characteristics. Generally metadata provides information such as end point references, service contact information, etc.

[2.](#) Protocol Transport

The metadata query protocol seeks to fully employ the features of the HTTP protocol. Additionally this specification makes mandatory some optional HTTP features.

[2.1.](#) Transport Protocol

The metadata query protocol makes use of the HTTP protocol to transmit requests and responses. The underlying HTTP connection may make use of any appropriate transport protocol. In particular, the HTTP connection MAY make use of either TCP or SSL/TLS at the transport layer. See the Security Considerations section for guidance in choosing an appropriate transport protocol.

[2.2.](#) HTTP Version

Requests from clients MUST NOT use an HTTP version prior to version

1.1. Responders MUST reply to such requests using status code 505, "HTTP Version Not Supported".

Protocol responders MUST support requests using HTTP version 1.1, and MAY support later versions.

[2.3.](#) HTTP Method

All metadata query requests MUST use the GET method.

[2.4.](#) Request Headers

Young

Expires July 2, 2014

[Page 3]

Internet-Draft

Metadata Query Protocol

December 2013

All metadata query requests MUST include the following HTTP headers:

Accept - this header MUST contain the content-type identifying the type, or form, of metadata to be retrieved

All metadata query requests SHOULD include the following HTTP headers:

Accept-Charset

Accept-Encoding

A metadata request to the same URL, after an initial request, MUST include the following header per [section 13.3.4 of RFC 2616](#) [RFC2616]:

If-None-Match

[2.5.](#) Response Headers

All successful metadata query responses (even those that return no results) MUST include the following headers:

Content-Encoding - required if, and only if, content is compressed

Content-Type

ETag

All metadata retrieval responses SHOULD include the following headers:

Cache-Control

Content-Length

Last-Modified

[2.6.](#) Status Codes

This protocol uses the following HTTP status codes:

200 "OK" - standard response code when returning requested metadata

304 "Not Modified" - response code indicating requested metadata has not been updated since the last request

Young

Expires July 2, 2014

[Page 4]

Internet-Draft

Metadata Query Protocol

December 2013

400 "Bad Request" - response code indicating that the requester's request was malformed in some fashion

401 "Unauthorized" - response code indicating the request must be authenticated before requesting metadata

404 "Not Found" - indicates that the requested metadata could not be found; this MUST NOT be used in order to indicate a general service error.

405 "Method Not Allowed" - response code indicating that a non-GET method was used

406 "Not Acceptable" - response code indicating that metadata is not available in the request content-type

501 "Not Implemented" - response code indicating that a given extended identifier is not supported

505 "HTTP Version Not Supported" - response code indicating that HTTP/1.1 was not used

[2.7.](#) Base URL

Requests defined in this document are performed by issuing an HTTP GET request to a particular URL. The final component of the path to which requests are issued is defined by the requests specified within this document. A base URL precedes such paths. Such a base URL **MUST** contain at least the scheme and host name components. It **MAY** also include a port as well as a path. It **MUST NOT** include URL fragments. If a path is included the path required by the particular defined request is appended to the path in the base URL.

[2.8.](#) Content Negotiation

As there may be many representations for a given piece of metadata, agent-driven content negotiation is used to ensure the proper representation is delivered to the requester. In addition to the required usage of the Accept header a responder **SHOULD** also support the use of the Accept-Charset header.

[3.](#) Metadata Query Protocol

The metadata query protocol retrieves metadata based on a single "tag" or "keyword" identifier. A request may return information for none, one, or a collection of entities.

[3.1.](#) Identifiers

The query protocol uses identifiers to "tag" metadata for single- and multi-entity metadata collections. The assignment of such identifiers to a particular metadata document is the responsibility of the query responder. If a metadata collection already contains a well known identifier it is **RECOMMENDED** that such a natural identifier is used when possible. Any given metadata collection **MAY** have more than one identifier associated with it.

An identifier used in the query protocol is either a basic identifier or an extended identifier:

id = basicid / extid

Identifiers starting with anything other than "{" are basic identifiers:

```
basicid  = notlb *idchar
notlb    = %x00-7a / %x7c-7f ; anything other than a "{"
```

An identifier starting with "{" (ASCII 0x7B) is an extended identifier, and MUST match the "extid" production in the following ABNF grammar:

```
extid    = "{" extname "}" extarg
extname  = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )
extarg   = *idchar
idchar   = %x00-ff ; any encodable character
ALPHA    = %x41-5a / %x61-7a
DIGIT    = %x30-39
```

The particular extension to be used is named by the "extname" value. This value is case sensitive; "Example" and "example" would name different extensions. Each extension may define its own additional rules for the "extarg" component.

A responder MUST return a 400 (bad request) status code if either of the following conditions hold:

- o An identifier starting with "{" does not match the "extid" production.
- o The "extarg" component fails to meet the requirements for the named extension.

A responder MUST return a 501 (not implemented) status code if the "extname" value is not supported by the responder.

[3.2.](#) Protocol

[3.2.1.](#) Request

A metadata query request is performed by issuing an HTTP GET request to a URL constructed as the concatenation of the following components:

- o The responder's base URL.
- o A single "/" character, unless the base URL already ends in a "/".
- o The string "entities/".
- o A single URL-encoded identifier.

For example, with a base URL of "http://example.org/mdq", a query for the identifier "foo" would be performed by an HTTP GET request to the following URL:

http://example.org/mdq/entities/foo

[3.2.2.](#) Response

The response to a metadata query request **MUST** be a document that provides metadata for the given request identifier in the format described by the request's Content-Type header.

The responder is responsible for ensuring that the metadata returned is valid. If the responder can not create a valid document it **MUST** respond with a 406 status code. An example of such an error would be the case where the result of the query is metadata for multiple entities but the request content type does not support returning multiple results in a single document.

[3.2.3.](#) Example Request and Response

```
GET /service/entities/http%3A%2F%2Fexample.org%2Fidp HTTP/1.1
Host: metadata.example.org
Accept: application/samlmetadata+xml
```

Example Metadata Query Request

Content-Type: application/samlmetadata+xml
ETag: abcdefg
Last-Modified: Thu, 15 Apr 2010 12:45:26 GMT
Content-Length: 1234

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor entityID="http://example.org/idp"
  xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
....
```

Example Metadata Query Response

[4.](#) Efficient Retrieval and Caching

[4.1.](#) Conditional Retrieval

Upon a successful response the responder is required to return an ETag header and may return a Last-Modified header as well. Requesters SHOULD use either or both, with the ETag being preferred, in any subsequent requests for the same resource. In the event that a resource has not changed since the previous request, the requester will receive a 304 (Not Modified) status code as a response.

[4.2.](#) Content Caching

Responders SHOULD include cache control information with successful (200 status code) responses, assuming the responder knows when retrieved metadata is meant to expire. The responder should also include cache control information with 404 Not Found responses. This allows the requester to create and maintain a negative-response cache. When cache controls are used only the 'max-age' directive SHOULD be used.

[4.3.](#) Content Compression

As should be apparent from the required request and response headers this protocol encourages the use of content compression. This is in recognition that some metadata documents can be quite large or fetched with relatively high frequency.

Requesters SHOULD support, and advertise support for, gzip compression unless such usage would put exceptional demands on constrained environments. Responders MUST support gzip compression. Requesters and responders MAY support other compression algorithms.

[5.](#) Protocol Extension Points

The Metadata Query Protocol is extensible using the following protocol extension points:

- o The "extension identifier" syntax described in the Identifiers section above is intended for relatively simple extensions such as transformed identifiers.
- o More complex extensions may be implemented by defining additional paths (other than "entities/") below the base URL.

[6.](#) Security Considerations

[6.1.](#) Integrity

As metadata often contains information necessary for the secure operation of interacting services it is RECOMMENDED that some form of content integrity checking be performed. This may include the use of SSL/TLS at the transport layer, digital signatures present within the metadata document, or any other such mechanism.

[6.2.](#) Confidentiality

In many cases service metadata is public information and therefore confidentiality is not required. In the cases where such functionality is required, it is RECOMMENDED that both the requester and responder support SSL/TLS. Other mechanisms, such as XML encryption, MAY also be supported.

[6.3.](#) Authentication

All responders which require client authentication to view retrieved information MUST support the use of HTTP basic authentication over SSL/TLS. Responders SHOULD also support the use of X.509 client certificate authentication.

[7.](#) IANA Considerations

This document has no actions for IANA.

[8.](#) Acknowledgements

The editor would like to acknowledge the following individuals for their contributions to this document:

Scott Cantor (The Ohio State University)

Internet-Draft

Metadata Query Protocol

December 2013

Thomas Lenggenhager (SWITCH)

Joe St Sauver (University of Oregon)

Tom Scavo (Internet2)

Special acknowledgement is due to Chad LaJoie (Covisint) for his work in editing previous versions of this specification.

[9.](#) Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [STD68] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.

[Appendix A.](#) Change Log (to be removed by RFC Editor before publication)

[A.1.](#) Since [draft-young-md-query-00](#)

Split into two documents: this one is as agnostic as possible around questions such as metadata format and higher level protocol use cases, a new layered document describes the detailed requirements for SAML support.

Rewrite [Section 3.2.1](#) to clarify construction of the request URL and its relationship to the base URL.

Added [Section 2.1](#) to clarify that the transport protocol underlying HTTP may be either TCP or SSL/TLS.

Clarify position on HTTP versions ([Section 2.2](#)) which may be used to underly this protocol.

Added Change Log modelled on [draft-ietf-httpbis-http2](#).

Added a reference to [STD68]. Use ABNF to describe request syntax. Replace transformed identifier concept with extended identifiers (this also resulted in the removal of any discussion of specific transformed identifier formats). Add grammar to distinguish basic from extended identifiers.

Young

Expires July 2, 2014

[Page 10]

Internet-Draft

Metadata Query Protocol

December 2013

Changed the required response when the result can not be validly expressed in the requested format from 500 to 406.

Removed the '+' operator and all references to multiple identifiers in queries. If more complex queries are required, these will be reintroduced at a different path under the base URL.

Added a section describing Protocol Extension Points.

[A.2](#). Since [draft-lajoie-md-query-01](#)

Adopted as base for [draft-young-md-query-00](#).

Updated author and list of contributors.

Changed ipr from "pre5378Trust200902" to "trust200902", submission type from IETF to independent and category from experimental to informational.

Added empty IANA considerations section.

Minor typographical nits but (intentionally) no substantive content changes.

Author's Address

Ian A. Young (editor)
Independent

EMail: ian@iay.org.uk

Young

Expires July 2, 2014

[Page 11]