Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 17, 2017

I. Young, Ed.
Independent

January 13, 2017

Metadata Query Protocol
draft-young-md-query-06

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.

This document is a product of the Research and Education Federations (REFEDS) Working Group process.

Editorial Note (To be removed by RFC Editor before publication)

Discussion of this draft takes place on the MDX mailing list (mdx@lists.iay.org.uk), which is accessed from [MDX.list].

XML versions, latest edits and the issues list for this document are available from [md-query].

The changes in this draft are summarized in Appendix A.7.

Status of This Memo

Table of Contents

1. Introduction 3
1.1. Notation and Conventions 3
1.2. Terminology 4
2. Protocol Transport 4
2.1. Transport Protocol 4
2.2. HTTP Version 4
2.3. HTTP Method 4
2.4. Request Headers 4
2.5. Response Headers 5
2.6. Status Codes 5
2.7. Base URL 6
2.8. Content Negotiation 6
3. Metadata Query Protocol 6
3.1. Identifiers 7
3.2. Protocol 7
3.2.1. Request by Identifier 7
3.2.2. Request All Entities 8
3.2.3. Response 8
3.2.4. Example Request and Response 8
4. Efficient Retrieval and Caching 9
4.1. Conditional Retrieval 9
4.2. Content Caching 9
4.3. Content Compression 9
5. Protocol Extension Points 10
6. Security Considerations 10
6.1. Integrity 10
6.2. Confidentiality 10
6.3. Authentication 10
7. IANA Considerations 10

8. Acknowledgements 10
9. References 11
9.1. Normative References 11
9.2. Informative References 12
Appendix A. Change Log (to be removed by RFC Editor before

publication) 13
A.1. Since draft-lajoie-md-query-01 13
A.2. Since draft-young-md-query-00 13
A.3. Since draft-young-md-query-01 14
A.4. Since draft-young-md-query-02 14
A.5. Since draft-young-md-query-03 14
A.6. Since draft-young-md-query-04 14
A.7. Since draft-young-md-query-05 15
Author's Address 15

1. Introduction

Many clients of web-based services are capable of consuming descriptive metadata about a service in order to customize or obtain information about 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 ([RFC7230] et seq.) 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.

The Research and Education Federations group ([REFEDS]) is the voice that articulates the mutual needs of research and education identity federations worldwide. It aims to represent the requirements of research and education in the ever-growing space of access and identity management.

From time to time REFEDS will wish to publish a document in the Internet RFC series. Such documents will be published as part of the RFC Independent Submission Stream [RFC4844]; however the REFEDS working group sign-off process will have been followed for these documents, as described in the REFEDS Participant's Agreement [REFEDS.agreement].

This document is a product of the REFEDS Working Group process.

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 ([RFC7230]) 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 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

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. See section 5.3.2 of [RFC7231].

All metadata query requests SHOULD include the following HTTP headers:

Accept-Charset, see section 5.3.3 of [RFC7231]

Accept-Encoding, see section 5.3.4 of [RFC7231]

A metadata request to the same URL, after an initial request, MUST include the following header:

If-None-Match, see section 3.2 of [RFC7232].

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. See section 3.1.2.2 of [RFC7231].

Content-Type, see section 3.1.1.5 of [RFC7231].

ETag, see section 2.3 of [RFC7232].

All metadata retrieval responses SHOULD include the following

headers:

Cache-Control, see section 5.2 of [RFC7234].

Content-Length, see section 3.3.2 of [RFC7230]

Last-Modified, see section 2.2 of [RFC7232].

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

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

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 ([STD66]). 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 the scheme and authority components.
- MUST contain a path component ending with a slash ('/') character.
- MUST NOT include a query component.
- MUST NOT include a fragment identifier component.

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 either for all entities known to the responder or for a named collection based on a single "tag" or "keyword" identifier. A request returns 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 a non-empty sequence of arbitrary 8-bit characters:

```
id       = 1*idchar
idchar   = %x00-ff ; any encodable character
```

3.2. Protocol

3.2.1. Request by Identifier

A metadata query request for all entities tagged with a particular identifier is performed by issuing an HTTP GET request to a URL constructed as the concatenation of the following components:

- The responder's base URL.
- The string "entities/".
- A single identifier, percent-encoded appropriately for use as a URL path segment (see sections 2.1 and 3.3 of [STD66]).

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

Correct encoding of the identifier as a URL path segment is critical for interoperability. In particular:

The character '/' MUST be percent-encoded.

The space character MUST be encoded as '%20' and MUST NOT be encoded as '+' as would be required in a query parameter.

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

http://example.org/mdq/entities/blue%2Fgreen+light%20blue

3.2.2. Request All Entities

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

- The responder's base URL.
- The string "entities".

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

http://example.org/mdq/entities

3.2.3. Response

The response to a metadata query request MUST be a document that provides metadata for the given request in the format described by the request's Accept 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.

3.2.4. Example Request and Response

The following example demonstrates a metadata query request using a base URL of "http://metadata.example.org/service/" and the identifier "http://example.org/idp".

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

Example Metadata Query Request

```
HTTP/1.x 200 OK
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 MUST 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 responder SHOULD send 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:

- Profiles of this specification may assign semantics to specific identifiers, or to identifiers structured in particular ways.

- Profiles of this specification may define additional paths (other than "entities" and "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 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 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 ([RFC7235], [RFC2617]/[I-D.basicauth]) over 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)

Leif Johansson (SUNET)

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. References

9.1. Normative References

[BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[I-D.basicauth] Reschke, J., "The 'Basic' HTTP Authentication Scheme", draft-ietf-httpauth-basicauth-update-07 (work in progress), February 2015.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

[RFC7232] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, June 2014.

[RFC7234] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

[RFC7235] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, June 2014.

[STD66] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[STD68] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

9.2. Informative References

[md-query]
Young, I., Ed., "md-query Project", <https://github.com/iay/md-query>.

[MDX.list]
Young, I., Ed., "MDX Mailing List", <http://lists.iay.org.uk/listinfo.cgi/mdx-iay.org.uk>.

[REFEDS] Research and Education Federations, "REFEDS Home Page", <http://www.refeds.org/>.

[REFEDS.agreement]
Research and Education Federations, "REFEDS Participant's Agreement", <https://refeds.org/about/about_agreement.html>.

[RFC4844] Daigle, L. and Internet Architecture Board, "The RFC Series and RFC Editor", RFC 4844, July 2007.

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

A.1. 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.

A.2. 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.

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.3. Since draft-young-md-query-01

Added REFEDS RFC stream boilerplate.

Tidied up some normative language.

A.4. Since draft-young-md-query-02

Introduced a normative reference to [STD66].

Reworked the definition of the base URL so that a non-empty path ending with '/' is required. This allows the definition of request URLs to be simplified.

Clarified the definition of the base URL to exclude a query component; corrected the terminology for the fragment identifier component.

Added the definition for the query for all entities in Section 3.2.2.

Corrected an example in Section 3.2.4 to include the required double quotes in the value of an ETag header. Added text to clarify the base URL and identifier being used in the example.

Simplified the definition of identifiers, so that any non-empty identifier is accepted and no semantics are defined for particular structures. Extended syntaxes such as the "{sha1}" notation for transformed identifiers are now left to profiles.

Remove incidental references to SSL.

Remove status code 501 ("not implemented") as it is no longer referenced.

A.5. Since draft-young-md-query-03

Correct a typo in the identifier grammar.

A.6. Since draft-young-md-query-04

Updated to rely on the new definition of HTTP/1.1 in [RFC7230] et seq. instead of RFC 2616.

Corrected Section 3.2.3 to indicate that the request contains an Accept header, not a Content-Type header.

Added an Editorial Note to help direct readers back to the discussion.

A.7. Since draft-young-md-query-05

Remove unnecessary percent-encoding of a ':' character in the example in Section 3.2.4.

Removed use of the ambiguous term "URL-encoded" in Section 3.2.1. Instead, indicate that the encoding must correspond to the rules for encoding a URL path segment specifically, and call out some of the more important implications arising from that. Added a new example illustrating these implications.

Updated the description of conditional retrieval in Section 4.1 to make the use of a 304 (Not Modified) status code a normative but non-mandatory obligation on the responder, not simply a description of what the requester will receive.

Author's Address

Ian A. Young (editor)
Independent

EMail: ian@iay.org.uk