Internet Engineering Task Force C. Ma
Internet Draft J. Chen
Intended status: Experimental X. Fan
Expires: June 25, 2020 M. Chen
Z. Li
China Academy of Information and Communications Technology
December 25, 2019

Industrial Internet Identifier Data Access Protocol (IIIDAP) Query Format
draft-mcd-identifier-access-query-00

Status of this Memo

Copyright Notice

Abstract

This document describes uniform patterns to construct HTTP URLs that may be used to retrieve identifier information from Second-Level Nodes (SLN) using "RESTful" web access patterns. These uniform patterns define the query syntax for the Industrial Internet Identifier Data Access Protocol (IIIDAP).

Table of Contents

1. Introduction

This document describes a specification for querying identifier data using a RESTful web service and uniform query patterns. The service is implemented using the Hypertext Transfer Protocol (HTTP) [RFC7230] and the conventions described in [IDENTIFIER-HTTP]. These uniform patterns define the query syntax for the Industrial Internet Identifier Data Access Protocol (IIIDAP).

The intent of the patterns described here are to enable queries of the identifier information by identifiers or names. [RFC3986] patterns specified in this document are only applicable to the HTTP [RFC7231] GET and HEAD methods. As described in Section 4.1 of [IDENTIFIER-HTTP], HEAD method can be used to determine, if an object exists (or not) without returning IIIDAP-encoded results; GET method can be used to retrieve detailed results.

This document does not describe the results or entities returned from issuing the described URLs with an HTTP GET. The specification of these entities is described in [IDENTIFIER-RESPONSES].

Additionally, resource management, provisioning, and update functions are out of scope for this document. Second-Level Nodes (SLN) have various and divergent methods covering these functions, and it is unlikely a uniform approach is needed for interoperability.

HTTP contains mechanisms for servers to authenticate clients and for clients to authenticate servers (from which authorization schemes may be built), so such mechanisms are not described in this document. Policy, provisioning, and processing of authentication and authorization are out of scope for this document as deployments will have to make choices based on local criteria. Supported authentication mechanisms are described in [IDENTIFIER-SECURITY].

2. Conventions used in this document

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 [RFC2119].

2.1. Acronyms and Abbreviations

TLN: Top-Level Nodes

SLN: Second-Level Nodes

ELN: Enterprise-Level Nodes

NFC: Unicode Normalization Form C [Unicode-UAX15]

NFKC: Unicode Normalization Form KC [Unicode-UAX15]

IIIDAP: Industrial Internet Identifier Data Access Protocol

REST: Representational State Transfer. The term was first described in a doctoral dissertation [REST].

RESTful: An adjective that describes a service using HTTP and the principles of REST.

3. Path Segment Specification

The base URLs used to construct IIIDAP queries are maintained in an TLN described in [IDENTIFIER-AUTHORIZATION]. Queries are formed by retrieving an appropriate base URL from the TLN and appending a path segment specified in either Sections 3.1 or 3.2. Generally, a TLN or other service provider will provide a base URL that identifies the protocol, host, and port, and this will be used as a base URL that the complete URL is resolved against, as per Section 5 of RFC 3986 [RFC3986]. For example, if the base URL is "https://example.com/iiidap/", all IIIDAP query URLs will begin with "https://example.com/iiidap/".

The bootstrap registry does not support searching for identifier information through query fields that are not part of a global namespace, including "name" and "help". A base URL for an associated object is required to construct a complete query.

3.1. Lookup Path Segment Specification

The resource type path segments for exact match lookup are:

- o 'identifier': Used to identify the identifier information of SLN or ELN query using a string identifier.

- o 'name': Used to identify the identifier information of SLN or ELN query using a node (SLN or ELN) name.

3.1.1. Identifier Path Segment Specification

Syntax: identifier/<Identifier of SLN or ELN>

Take Handle Protocol [RFC3651] as an example, Identifier format of TLN is XX; identifier format of SLD is XX.YY; identifier format of ELD is XX.YY.ZZ; XX, YY, ZZ are UTF-8 [6] encoded character strings, which use any characters from the Unicode 2.0 standard except the ASCII character '/' (0x2F). Therefore, queries for information about identifiers are of the form /identifier/XX.YY/...(for SLN) or /identifier/XX.YY.ZZZ/...(for ELN)

For example, identifier of a TLN can be 86; identifier of a SLN can be 86.100; identifier of a TLN can be 86.100.1; Identifiers of TLN/SLN/ELN are usually called identifiers prefix in Industry Internet Identifier System. Identifier suffix is used to identify a product or component in an enterprise. Prefixes and suffixes are joined by backslashes. The prefix and suffix together constitute the globally qualified identity for a product or component. Naming rules of identifier suffix is beyond the scope of this specification. The length of an identifier of must range from 2 to 255 number characters.

For example, the following URL would be used to find identifier information for the most specific identifier of a SLN containing 86.100:

```
https://example.com/iiidap/identifier/86.100
```

The following URL would be used to find identifier information for the most specific identifier of an ELN containing 86.100.1:

```
https://example.com/iiidap/identifier/86.100.1
```

3.1.2. Name Path Segment Specification

Syntax: name/<name of SLN or ELN>

Queries for identifier information regarding name of SLN or ELN are of the form /name/XXX/... where XXX is the name of a SLN or ELN. XXX is a string of length from 1 to 255. It can contain non-US-ASCII characters. The detailed requirements for character encoding are specified in Section 5.1.

For example, the following URL would be used to find identifier information of the node named "mengniu":

```
https://example.com/iiidap/name/mengniu
```

3.1.3. Help Path Segment Specification

Syntax: help

The help path segment can be used to request helpful information (command syntax, terms of service, privacy policy, rate-limiting policy, supported authentication methods, supported extensions, technical support contact, etc.) from an IIIDAP server. The response to "help" should provide basic information that a client needs to

successfully use the service. The following URL would be used to return "help" information:

```
https://example.com/iiidap/help
```

3.2. Search Path Segment Specification

Pattern matching semantics are described in Section 4.1. The name path segment for search is:

- 'names': Used to identify a SLN or ELN identifier information search using a pattern to match a fully qualified SLN/ELN name.

IIIDAP search path segments are formed using a concatenation of the plural form of the object being searched for and an HTTP query string. The HTTP query string is formed using a concatenation of the question mark character ('?', US-ASCII value 0x003F), the JSON object value associated with the object being searched for, the equal sign character ('=', US-ASCII value 0x003D), and the search pattern. Search pattern query processing is described more fully in Section 4. For the name described in this document, the plural object forms are "names".

3.2.1. Name Search

Syntax: names?name=<name search pattern>

Searches for identifier information by name are specified using this form:

names?name=XXXX

XXXX is a search pattern representing a name of SLN OR ELN. The following URL would be used to find identifier information for SLN or ELN names matching the "example*" pattern:

```
https://example.com/iiidap/names?name=example*
```

4. Query Processing

Servers indicate the success or failure of query processing by returning an appropriate HTTP response code to the client. Response codes not specifically identified in this document are described in [IDENTIFIER-HTTP].

4.1. Partial String Searching

Partial string searching uses the asterisk ('*', US-ASCII value 0x002A) character to match zero or more trailing characters. A character string representing multiple names MAY be concatenated to the end of the search pattern to limit the scope of the search. For example, the search pattern "exam*" will match "example1" and "example2". The search pattern "ex*mple" will match "example". If an asterisk appears in a search string, any label that contains the non-asterisk characters in sequence plus zero or more characters in sequence in place of the asterisk would match. Additional pattern matching processing is beyond the scope of this specification.

If a server receives a search request but cannot process the request because it does not support a particular style of partial match searching, it SHOULD return an HTTP 422 (Unprocessable Entity) [RFC4918] response. When returning a 422 error, the server MAY also return an error response body as specified in Section 6 of [IDENTIFIER-RESPONSES] if the requested media type is one that is specified in [IDENTIFIER-HTTP].

Partial matching is not feasible across combinations of Unicode characters because Unicode characters can be combined with each other. Servers SHOULD NOT partially match combinations of Unicode characters where a legal combination is possible. It should be noted, though, that it may not always be possible to detect cases where a character could have been combined with another character, but was not, because characters can be combined in many different ways.

Clients should avoid submitting a partial match search of Unicode characters where a Unicode character may be legally combined with another Unicode character or characters. Partial match searches with incomplete combinations of characters where a character must be combined with another character or characters are invalid. Partial match searches with characters that may be combined with another character or characters are to be considered non-combined characters (that is, if character x may be combined with character y but character y is not submitted in the search string, then character x is a complete character and no combinations of character x are to be searched).

4.2. Associated Records

Conceptually, any query-matching record in a server's database might be a member of a set of related records, related in some fashion as defined by the server. The entire set ought to be considered as

candidates for inclusion when constructing the response. However, the construction of the final response needs to be mindful of privacy and other data-releasing policies when assembling the IIIDAP response set.

Note too that due to the nature of searching, there may be a list of query-matching records. Each one of those is subject to being a member of a set as described in the previous paragraph. What is ultimately returned in a response will be the union of all the sets that has been filtered by whatever policies are in place.

Note that this model includes arrangements for associated names, including those that are linked by policy mechanisms and names bound together for some other purposes. Note also that returning information that was not explicitly selected by an exact-match lookup, including additional names that match a relatively fuzzy search as well as lists of names that are linked together, may cause privacy issues.

Note that there might not be a single, static information return policy that applies to all clients equally. Client identity and associated authorizations can be a relevant factor in determining how broad the response set will be for any particular query.

5. Internationalization Considerations

5.1. Character Encoding Considerations

Servers can expect to receive search patterns from clients that contain character strings encoded in different forms supported by HTTP. It is entirely possible to apply filters and normalization rules to search patterns prior to making character comparisons, but this type of processing is more typically needed to determine the validity of registered strings than to match patterns.

An IIIDAP client submitting a query string containing non-US-ASCII characters converts such strings into Unicode in UTF-8 encoding. It then performs any local case mapping deemed necessary. Strings are normalized using Normalization Form C (NFC) [Unicode-UAX15]; note that clients might not be able to do this reliably. UTF-8 encoded strings are then appropriately percent-encoded [RFC3986] in the query URL.

After parsing any percent-encoding, an IIIDAP server treats each query string as Unicode in UTF-8 encoding. If a string is not valid UTF-8, the server can immediately stop processing the query and return an HTTP 400 (Bad Request) response.

For everything else, servers map fullwidth and halfwidth characters to their decomposition equivalents. Servers convert strings to the same coded character set of the target data that is to be looked up or searched, and each string is normalized using the same normalization that was used on the target data. In general, storage of strings as Unicode is RECOMMENDED. For the purposes of comparison, Normalization Form KC (NFKC) [Unicode-UAX15] with case folding is used to maximize predictability and the number of matches. Note the use of case-folded NFKC as opposed to NFC in this case.

6. Security Considerations

Security services for the operations specified in this document are described in "Security Services for the Industrial Internet Identifier Data Access Protocol (IIIDAP)" [IDENTIFIER-SECURITY].

Search functionality typically requires more server resources (such as memory, CPU cycles, and network bandwidth) when compared to basic lookup functionality. This increases the risk of server resource exhaustion and subsequent denial of service due to abuse. This risk can be mitigated by developing and implementing controls to restrict search functionality to identified and authorized clients. If those clients behave badly, their search privileges can be suspended or revoked. Rate limiting as described in Section 5.5 of "HTTP Usage in the Industrial Internet Identifier Data Access Protocol (IIIDAP)" [IDENTIFIER-HTTP] can also be used to control the rate of received search requests. Server operators can also reduce their risk by restricting the amount of information returned in response to a search request.

Search functionality also increases the privacy risk of disclosing object relationships that might not otherwise be obvious.

Note that there might not be a single, static information return policy that applies to all clients equally. Client identity and associated authorizations can be a relevant factor in determining how broad the response set will be for any particular query.

7. IANA Considerations

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

[RFC4918] Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007, <http://www.rfc-editor.org/info/rfc4918>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

[Unicode-UAX15]
The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", September 2013, <http://www.unicode.org/reports/tr15/>.

8.2. Informative References

[RFC3651] Sun, S., Reilly, S. and L. Lannom, "Handle System Namespace and Service Definition", RFC 3651, November 2003, <http://www.rfc-editor.org/info/rfc3651>.

[REST] Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. Dissertation, University of California, Irvine, 2000, <http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf>.

[IDENTIFIER-HTTP]
Ma, C., "HTTP Usage in the Industrial Internet Identifier Data Access Protocol (IIIDAP)", Work in Progress, draft-ma-identifier-access-http-00, December 2019.

[IDENTIFIER-SECURITY]
Ma, C., "Security Services for the Industrial Internet Identifier Data Access Protocol (IIIDAP)", Work in Progress, draft-mcd-identifier-access-security-00, December 2019.

[IDENTIFIER-RESPONSES]
Ma, C., "JSON Responses for the Industrial Internet Identifier Data Access Protocol (IIIDAP)", Work in Progress, draft-mcd-identifier-access-responce-00, December 2019.

[IDENTIFIER-AUTHORIZATION]
Ma, C., "Finding the Authoritative Industrial Internet Identifier Data (IIIDAP) Service", Work in Progress, draft-mcd-identifier-access-authority-00, December 2019.

Authors' Addresses

Chendi Ma
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 177 1090 9864
Email: machendi@caict.ac.cn

Chen Jian
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 138 1103 3332
Email: chenjian3@caict.ac.cn

Xiaotian Fan
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 134 0108 6945
Email: fanxiaotian@caict.ac.cn

Meilan Chen
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 139 1143 7301
Email: chenmeilan@caict.ac.cn

Zhiping Li
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 185 1107 1386
Email: lizhiping@caict.ac.cn