

Internet Engineering Task Force
Internet Draft
Intended status: Experimental
Expires: June 25, 2020

C. Ma
J. Chen
X. Fan
M. Chen
Z. Li

China Academy of Information and Communications Technology
December 25, 2019

**JSON Responses for the Industrial Internet Identifier Data Access
Protocol (IIIDAP)
draft-mcd-identifier-access-responce-00**

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on June 25, 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 (<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. 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.

Abstract

This document describes JSON data structures representing identifier information maintained by Second-Level Nodes (SLN). These data structures are used to form Industrial Internet Identifier Data Access Protocol (IIIDAP) query responses.

Table of Contents

1.	Introduction	2
1.1.	Terminology and Definitions.....	3
1.2.	Data Model	3
2.	Use of JSON	4
2.1.	Naming	4
3.	Common Data Types	4
4.	Common Data Structures.....	5
4.1.	IIIDAP Conformance.....	5
4.2.	Links	5
4.3.	Notices and Remarks.....	6
4.4.	Language Identifier.....	8
5.	Nodes	8
6.	Error Response Body	12
7.	Responding to Help Queries.....	14
8.	Responding To Searches.....	14
9.	Indicating Truncated Responses.....	15
10.	IANA Considerations.....	17
10.1.	IIIDAP JSON Media Type Registration.....	17
11.	Security Considerations.....	18
12.	Internationalization Considerations.....	18
12.1.	Character Encoding.....	18
12.2.	URIs and IRIs	18
13.	Privacy Considerations.....	18
14.	References	19
14.1.	Normative References.....	19
14.2.	Informative References.....	20

[1. Introduction](#)

This document describes responses in the JSON [[RFC7159](#)] format for the queries as defined by the Industrial Internet Identifier Data Access Protocol Query Format [[IDENTIFIER-QUERY](#)]. A communication

protocol for exchanging queries and responses is described in [\[IDENTIFIER-HTTP\]](#).

[1.1. Terminology and Definitions](#)

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](#) [[RFC2119](#)].

The following list describes terminology and definitions used throughout this document:

TLN: Top-Level Nodes

SLN: Second-Level Nodes

ELN: Enterprise-Level Nodes

member: data found within an object as defined by JSON [[RFC7159](#)].

object: a data structure as defined by JSON [[RFC7159](#)].

IIIDAP: Industrial Internet Identifier Data Access Protocol

[1.2. Data Model](#)

The data model for JSON responses is specified in five sections:

1. simple data types conveyed in JSON strings
2. data structures specified as JSON arrays or objects that are used repeatedly when building up larger objects
3. arrays of objects representing structured data corresponding to a search for multiple objects
4. the response to an error

Positive responses take two forms. A response to a lookup of a single object yields a JSON object, which is the subject of the lookup. A response to a search for multiple objects yields a JSON object that contains an array of JSON objects that are the subject of the search. In each type of response, other data structures are present within the topmost JSON object.

2. Use of JSON

2.1. Naming

Clients of these JSON responses SHOULD ignore unrecognized JSON members in responses.

Clients processing JSON responses need to be prepared for members representing identifier data specified in this document to be absent from a response. In other words, servers are free to not include JSON members containing identifier data based on their own policies.

Finally, all JSON names specified in this document are case sensitive. Both servers and clients MUST transmit and process them using the specified character case.

3. Common Data Types

JSON [[RFC7159](#)] defines the data types of a number, character string, boolean, array, object, and null. This section describes the semantics and/or syntax reference for common, JSON character strings used in this document.

identifier: 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).

name: The maximum limit of name field is 255 ASCII characters.

picture: Pictures need to be Base64 encoded, then transferred via json.

id: The maximum limit of id field is 32 ASCII characters.

dates and times: The syntax for values denoting dates and times is defined in [[RFC3339](#)].

URIs: The syntax for values denoting a Uniform Resource Identifier (URI) is defined by [[RFC3986](#)].

4. Common Data Structures

This section defines common data structures used in responses.

4.1. IIIDAP Conformance

The data structure named "iiidapConformance" is an array of strings, each providing a hint as to the specifications used in the construction of the response. This data structure appears only in the topmost JSON object of a response.

An example iiidapConformance data structure:

```
"iiidapConformance" :  
  
[  
  
  "iiidap_level_0"  
  
]
```

Figure 1

The string literal "iiidap_level_0" signifies conformance with this specification.

4.2. Links

The "links" array is found in data structures to signify links to other resources on the Internet. The relationship of these links is defined by the IANA registry described by [[RFC5988](#)].

The following is an example of the link structure:

```
{  
  "value" : "http://example.com/context_uri",  
  "rel" : "self",  
  "href" : "http://example.com/target_uri",  
  "hreflang" : [ "en", "ch" ],  
  "title" : "title",  
  "media" : "screen",  
  "type" : "application/json"  
}
```

Figure 2

The JSON name/values of "rel", "href", "hreflang", "title", "media", and "type" correspond to values found in [Section 5 of \[RFC5988\]](#). The "value" JSON value is the context URI as described by [\[RFC5988\]](#). The "href" JSON value MUST be specified. All other JSON values are OPTIONAL.

This is an example of the "links" array as it might be found in an notice:

```
"links" :  
[  
  {  
    "href" : "http://example.com",  
    "title" : ""Terms of Use""  
  }  
]
```

Figure 3

[4.3.](#) Notices and Remarks

The "notices" and "remarks" data structures take the same form. The notices structure denotes information about the service providing IIIDAP information and/or information about the entire response, whereas the remarks structure denotes information about the identifiers that contains it.

Both are arrays of objects. Each object contains an optional "title" string representing the title of the object, an optional "type" string denoting a registered type of remark or notice, an array of strings named "description" for the purposes of conveying any descriptive text, and an optional "links" array as described in [Section 4.2](#).

An example of the notices data structure:

```
"notices" :  
[  
  {  
    "title" : "Terms of Use",
```



```
"description" :
[
  "Service subject to The Registry of the Moon's TOS.",
  "Copyright (c) 2020 LunarNIC"
],
"links" :
[
  {
    "value" : "http://example.net/entity/XXXX",
    "rel" : "alternate",
    "type" : "text/html",
    "href" : http://www.example.com/terms_of_use.html
  }
]
}
```

Figure 4

It is the job of the clients to determine line breaks, spacing, and display issues for sentences within the character strings of the "description" array. Each string in the "description" array contains a single complete division of human-readable text indicating to clients where there are semantic breaks.

An example of the remarks data structure:

```
"remarks" :
[
  {
    "description" :
    [
      "She sells sea shells down by the sea shore.",
      "Originally written by Terry Sullivan."
    ]
  }
]
```

Figure 5

Note that objects in the "remarks" array may also have a "links" array.

While the "title" and "description" fields are intended primarily for human consumption, the "type" string contains a well-known value to be registered with TLN for programmatic use.

An example of the remarks data structure:

```
"remarks" :
[
  {
    "type" : "information truncated due to authorization",
    "description" :
    [
      "Some identifier data may not have been given.",
      "Use proper authorization credentials to see all of it."
    ]
  }
]
```

Figure 6

4.4. Language Identifier

This data structure consists solely of a name/value pair, where the name is "lang" and the value is a string containing a language identifier as described in [\[RFC5646\]](#).

```
"lang" : "mn-Cyrl-MN"
```

5. Nodes

An identifier appears throughout this document and is an appropriate response for the /identifier/XXXX query defined in "Industrial Internet Identifier Data Access Protocol (IIIDAP) Query Format" [\[IDENTIFIER-QUERY\]](#). It represents the information of a SLN or ELN. All of these representations are expected to represent in JSON [\[RFC7159\]](#).

The following is an example of an entity.

```
{
  "iiidapConformance" :
  [
    "iiidap_level_0"
  ],
  "notices" :
  [
    {
      "title" : "Content Removed",
      "description" :
      [
        "Without full authorization, content has been removed.",

```



```
    "Sorry, dude!"
  ],
  "links" :
  [
    {
      "value" : "http://example.net/identifier/86.100.1",
      "rel" : "alternate",
      "type" : "text/html",
      "href" : http://www.example.com/redaction_policy.html
    }
  ]
},
"lang" : "en",
"identifier" : "86.100.1",
"regTime" : "2017-09-12T08:57:32.0Z",
"status" : "ok",
"node" :
{
  "name" : "mengniu"
  "nature" : "research institutions",
  "type" : "education",
  "address" :
    "No.52 Huayuan North Road, Haidian District Beijing, Beijing",
  "idNumber" : "84929141111234",
  "idType" : "other",
  "idPhoto" :
  {
    "mime": "data:image/jpeg;base64",
    "data":
"/9j/4AAaebZJRgABAQEASABIAAD/ha6aYeDr18DWornKr1X+/sP/9k="
  },
  "introduction" : "hello, i am ok",
  "legalPerson" :
  [
    {
      "name" : "San Zhang",
      "idNumber" : "131187199507297827",
      "idType" : "resident ID card",
      "idPhoto" :
      {
        "mime": "data:image/jpeg;base64",
        "data":
"/9j/4AAaebZJRgABAQEASABIAAD/ha6aYeDr18DWornKr1X+/sP/
9k="
      }
    }
  ]
}
```



```
"contact" :
[
  {
    "name" : "Si Li",
    "phone" : "+86 100 0138 5070",
    "email" : test@caict.ac.cn
  }
],
"remarks" :
[
  {
    "description" :
    [
      "She sells sea shells down by the sea shore.",
      "Originally written by Terry Sullivan."
    ]
  }
]
}
```

Figure 7

An identifier of a SLN or ELN can contain the following members:

- o identifier -- a unique identifier of a SLN or ELN
- o regTime -- a string representing the registration time of the identifier of a node
- o status -- a string representing the registration status of the identifier. Optional values of the status MUST be defined by the TLN. For example, status of nodes can be classified as: "ok" and "hold".
- o node -- an object representing a SLN or ELN, each containing the following values:
 - * name -- a sting containing the name of a node
 - * nature -- a string representing the nature of a node. The nature of a node (SLN or ELN) MUST be defined by the TLN. For example, according to their nature, nodes can be classified as: "government organs", "research institutions", "social organizations", and "enterprises and institutions".

- * `type` -- a string containing the type of industry the node belongs to. The type of a node (SLN or ELN) MUST be defined by the TLN. For example, types of nodes can be classified as:
 - "agriculture, forestry, animal husbandry and fishery",
 - "the mining industry",
 - "manufacturing",
 - "electricity, heat, gas and water production and supply",
 - "the construction industry",
 - "wholesale and retail",
 - "transportation, warehousing and postal services",
 - "accommodation and catering",
 - "information transmission, software and information technology services"
 - "the financial sector",
 - "the real estate industry",
 - "leasing and business services",
 - "scientific research and technology services",
 - "water conservancy, environment and public facilities management",
 - "residential services, repairs and other services",
 - "education",
 - "health and social work",
 - "culture, sports and entertainment",
 - "public administration, social security and social organization",
 - and "the international organization".
- * `address` -- a string containing the address of the node
- * `idNumber` -- a string representing the unique legally binding identification number of a node
- * `idType` -- a string representing the type of the `idNumber` of the node. The type of the identification number of a node (SLN or ELN) MUST be defined by the TLN. For example, according to their type, identification number of a node can be classified as: "unified social credit code", and "other".
- * `idPhoto` -- an object containing the picture messages for the legally binding entity proof of a node. The file type of the image can be .jpg, .png, .bmf and .jpeg, and base64 encoding is required for the file upload.
- * `introduce` -- a string describing some business information of the node

- * legalPerson -- an object representing the legal person of the node
 - + name -- a string containing the name of the legal person
 - + idNumber -- a string representing the unique legally binding identification number of the legal person
 - + idType -- a string representing the type of the idNumber of the legal person. The type of the identification number of a contact MUST be defined by the TLN. For example, according to their type, identification number of a contact can be classified as:
 - "Chinese identity card", "Travel permit for Hong Kong and Macao residents to and from the mainland",
 - "Permit for Taiwan residents to travel to and from the mainland",
 - "Permanent resident identity CARDS for foreigners",
 - "Hong Kong, Macao and Taiwan resident permit", and
 - "passport".
 - + idPhoto -- an object containing the picture messages for the legally binding entity proof of a legal person. The file type of the image can be .jpg, .png, .bmf and .jpeg, and base64 encoding is required for the file upload.
- * contact -- an object representing the of the node
 - + name -- a string containing the name of the contact
 - + phone -- a string containing the phone of the contact
 - + email -- a string containing the Email of the contact

6. Error Response Body

Some non-answer responses may return entity bodies with information that could be more descriptive.

The basic structure of that response is an object containing an error code number (corresponding to the HTTP response code) followed by a string named "title" and an array of strings named "description".

This is an example of the common response body.

```
{  
  "errorCode": 418,
```



```
"title": "Your Beverage Choice is Not Available",
"description":
[
  "I know coffee has more umpppphhh.",
  "Sorry, dude!"
]
}
```

Figure 8

This is an example of the common response body with an `iiidapConformance` and `notices` data structures:

```
{
  "iiidapConformance" :
  [
    "iiidap_level_0"
  ],
  "notices" :
  [
    {
      "title" : "Beverage Policy",
      "description" :
      [
        "Beverages with caffeine for keeping horses awake."
      ],
      "links" :
      [
        {
          "value" : "http://example.net/identifier/86.100.1",
          "rel" : "alternate",
          "type" : "text/html",
          "href" : "http://www.example.com/redaction_policy.html"
        }
      ]
    }
  ],
  "lang" : "en",
  "errorCode": 418,
  "title": "Your beverage choice is not available",
  "description":
  [
    "I know coffee has more umpppphhh.",
    "Sorry, dude!"
  ]
}
```


Figure 9

7. Responding to Help Queries

The appropriate response to /help queries as defined by [IDENTIFIER-QUERY] is to use the notices structure as defined in [Section 4.3](#).

This is an example of a response to a /help query including the iiidapConformance data structure.

```
{
  "iiidapConformance" :
  [
    "iiidap_level_0"
  ],
  "notices" :
  [
    {
      "title" : "Authentication Policy",
      "description" :
      [
        "Access to sensitive data for users with proper
credentials."
      ],
      "links" :
      [
        {
          "value" : "http://example.net/help",
          "rel" : "alternate",
          "type" : "text/html",
          "href" : "http://www.example.com/auth_policy.html"
        }
      ]
    }
  ]
}
```

Figure 10

8. Responding To Searches

[IDENTIFIER-QUERY] specifies the pattern of searches through the names of SLN and ELN. Responses to searches take the form of an array of identifier objects. These arrays are contained within the response object.

The following is an elided example of a response to a /names search.


```

{
  "iiidapConformance" :
  [
    "iiidap_level_0"
  ],
  ...
  "NodeSearchResults" :
  [
    {
      "identifier" : "86.100.1",
      "regTime" : "2019-12-12T08:57:32.0Z",
      "status" : "ok",
      ...
    },
    {
      "identifier" : "86.100.2",
      "regTime" : "2018-12-12T08:57:32.0Z",
      "status" : "ok",
      ...
    }
  ]
}

```

Figure 11

9. Indicating Truncated Responses

In cases where the data of a response needs to be limited or parts of the data need to be omitted, the response is considered "truncated". A truncated response is still valid JSON, but some of the results in a search set or some of the data in an object are not provided by the server. A server may indicate this by including a typed notice in the response object.

```

{
  "iiidapConformance" :
  [
    "iiidap_level_0"
  ],
  "notices" :
  [
    {
      "title" : "Search Policy",
      "type" : "result set truncated due to authorization",
      "description" :
      [
        "Search results are limited to 25 per day per querying IP."
      ]
    }
  ]
}

```



```

    ],
    "links" :
    [
      {
        "value" : "http://example.net/help",
        "rel" : "alternate",
        "type" : "text/html",
        "href" : "http://www.example.com/search_policy.html"
      }
    ]
  }
],
"domainSearchResults" :
[
  ...
]
}

```

Figure 12

A similar technique can be used with a typed remark where a single object has been returned and data in that object has been truncated.

The following is an elided example of a truncated data.

```

{
  "identifier" : "86.100.1",
  "regTime" : "2019-12-12T08:57:32.0Z",
  "status" : "ok",
  "node" :
    {
      "name" : "",
      "nature" : "research institutions",
      "type" : "education",
      ...
    }
  "legalPerson" :
  {
    ...
  }
  "contact" :
  {
    ...
  }
  },
  "remarks" :

```



```
[
  {
    "title" : "Data Policy",
    "type" : "object truncated due to unexplainable reason",
    "description" :
      [
        "Some of the data in this object has been removed."
      ],
    "links" :
      [
        {
          "value" : "http://example.net/help",
          "rel" : "alternate",
          "type" : "text/html",
          "href" : "http://www.example.com/data_policy.html"
        }
      ]
  }
]
```

Figure 13

10. IANA Considerations

10.1. IIIDAP JSON Media Type Registration

This specification will register the "application/iiidap+json" media type.

Type name: application

Subtype name: iiidap+json

Required parameters: n/a

Encoding considerations: See [Section 3.1 of \[RFC6839\]](#).

Security considerations: The media represented by this identifier does not have security considerations beyond that found in [Section 6 of \[RFC7159\]](#).

Interoperability considerations: There are no known interoperability problems regarding this media format.

Work in progress specification: [draft-mcd-identifier-access-responce-00](#)

Applications that use this media type: Implementations of the Industrial Internet Identifier Data Access Protocol (IIIDAP).

Intended usage: COMMON

Restrictions on usage: none

Author: Chendi Ma

Change controller: IETF

Provisional Registration: No (upon publication of this draft)

[11. Security Considerations](#)

This specification model information serialized in JSON format. As JSON is a subset of JavaScript, implementations are advised to follow the security considerations outlined in [Section 6 of \[RFC7159\]](#) to prevent code injection.

Though not specific to JSON, IIIDAP implementers should be aware of the security considerations specified in [\[IDENTIFIER-HTTP\]](#) and the security requirements and considerations in [\[IDENTIFIER-SECURITY\]](#).

Finally, service operators should be aware of the privacy mechanisms noted in [Section 13](#).

[12. Internationalization Considerations](#)

[12.1. Character Encoding](#)

The default text encoding for JSON responses in IIIDAP is UTF-8 [\[RFC3629\]](#), and all servers and clients MUST support UTF-8.

[12.2. URIs and IRIs](#)

[\[IDENTIFIER-HTTP\]](#) defines the use of URIs and IRIs in IIIDAP.

[13. Privacy Considerations](#)

This specification suggests some of values to identifier information that has been marked as private and/or has been removed or obscured, except:

- o identifier
- o name of a node
- o type of a node
- o introduce of a node

A few of the status values indicate that there are privacy concerns associated with the identifier object. The following status codes SHOULD be used to describe data elements of a response when appropriate:

- o private -- The object is not be shared in query responses, unless the user is authorized to view this information.
- o removed -- Data elements within the object have been collected but have been omitted from the response. This option can be used to prevent unauthorized access to associated identifier objects without the need to mark them as private.
- o obscured -- Data elements within the object have been collected, but the response value has been altered so that values are not easily discernible. A value changed from "1212" to "XXXX" is an example of obscured data. This option may reveal privacy sensitive information and should only be used when data sensitivity does not require a more protective option like "private" or "removed".

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), July 2002, <http://www.rfc-editor.org/info/rfc3339>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003, <http://www.rfc-editor.org/info/rfc3629>.

- [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>>.
- [RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", [BCP 47](#), [RFC 5646](#), September 2009, <<http://www.rfc-editor.org/info/rfc5646>>.
- [RFC5988] Nottingham, M., "Web Linking", [RFC 5988](#), October 2010, <<http://www.rfc-editor.org/info/rfc5988>>.
- [RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", [RFC 7159](#), March 2014, <<http://www.rfc-editor.org/info/rfc7159>>.

14.2. Informative References

- [JSON_ascendancy] MacVittie, L., "The Stealthy Ascendancy of JSON", April 2011, <<https://devcentral.f5.com/weblogs/macvittie/archive/2011/04/27/the-stealthy-ascendancy-of-json.aspx>>.
- [JSON_performance_study] Nurseitov, N., Paulson, M., Reynolds, R., and C. Izurieta, "Comparison of JSON and XML Data Interchange Formats: A Case Study", 2009, <<http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), August 2009, <<http://www.rfc-editor.org/info/rfc5730>>.
- [RFC6839] Hansen, T. and A. Melnikov, "Additional Media Type Structured Syntax Suffixes", [RFC 6839](#), January 2013, <<http://www.rfc-editor.org/info/rfc6839>>.
- [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-QUERY]

Ma, C., "Industrial Internet Identifier Data Access
Protocol (IIIDAP) Query Format", Work in Progress, [draft-
mcd-identifier-access-query-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