

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2012

N. Kong
L. Zhou
J. Xie
S. Shen
CNNIC
June 3, 2012

Domain Name Registration Data Access Protocol Response Format defined in
JSON
[draft-kong-dnrd-ap-response-json-00](#)

Abstract

This document describes a response format for registration data (Whois data) through RESTful Web-based Service. This response format is defined in JavaScript Object Notation (JSON).

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 December 5, 2012.

Copyright Notice

Copyright (c) 2012 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

- [1. Introduction](#) [3](#)
- [2. Convention Used in This Document](#) [3](#)
- [2.1. Abbreviations and Terminology](#) [3](#)
- [3. Response Format Specification](#) [4](#)
- [3.1. Domain Name Response Specification](#) [4](#)
- [3.1.1. Objects in the Domain Name Response](#) [5](#)
- [3.1.2. Extension Interface](#) [7](#)
- [3.1.3. JSON Format](#) [7](#)
- [3.2. Nameserver Response Specification](#) [9](#)
- [3.2.1. Objects in the Nameserver Response](#) [9](#)
- [3.2.2. Extension Interface](#) [10](#)
- [3.2.3. JSON Format](#) [10](#)
- [3.3. Contact Response Specification](#) [11](#)
- [3.3.1. Objects in the Contact Response](#) [12](#)
- [3.3.2. Extension Interface](#) [13](#)
- [3.3.3. JSON Format](#) [13](#)
- [3.4. Registrar Response Specification](#) [14](#)
- [3.4.1. Objects in the Registrar Response](#) [15](#)
- [3.4.2. Extension Interface](#) [16](#)
- [3.4.3. JSON Format](#) [17](#)
- [4. Accept Header](#) [19](#)
- [5. Internationalization Considerations](#) [19](#)
- [5.1. Encoding Specification](#) [19](#)
- [5.2. IDN Variants](#) [19](#)
- [6. IANA Considerations](#) [19](#)
- [7. Security considerations](#) [19](#)
- [8. Acknowledgements](#) [19](#)
- [9. References](#) [20](#)
- [9.1. Normative References](#) [20](#)
- [9.2. Informative References](#) [20](#)
- [Authors' Addresses](#) [21](#)

1. Introduction

Internet registries for both number resources and names have historically maintained a lookup service to permit public access to some portion of the registry database. Most registries offer the service via the WHOIS protocol [[RFC3912](#)], with additional services being offered via world wide web pages, bulk downloads, and other services, such as RPSL [[RFC2622](#)].

Although the WHOIS protocol specified in [[RFC3912](#)] is widely adopted and supported, it has several shortcomings that limits its usefulness to the evolving needs of the Internet community. For example, the WHOIS protocol has not been Internationalized, it does not consistently support Internationalized Domain Name (IDN, described in [[RFC5890](#)]); WHOIS has no query and response format; and WHOIS protocol does not support user authentication, access control for differentiated access.

Recently, the Internet Engineering Task Force (IETF) chartered a new working group to standardize a web-based (RESTful) Extensible Registration Data Access Protocol. This document, to be read in conjunction with the RESTful Query draft [[draft-hollenbeck-dnrd-ap-query-01](#)], describes the response specification for querying registration data through RESTful web service.

The response formats for querying domain, contact, nameserver name and registrar are defined in JSON, which is specified based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 ([[ECMA-262](#)]) and [[RFC4627](#)].

2. Convention 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. Abbreviations and Terminology

Domain Name Registration Data (DNRD) - refers to the information that registrants provide when registering a domain name. Registrars and registries collect this information and typically make a subset available to the public via the Whois service.

JavaScript Object Notation (JSON) is a lightweight data-interchange format. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 ([[ECMA-262](#)])

and [\[RFC4627\]](#).

Representational state transfer (REST) is a software architecture style for distributed systems such as the World Wide Web (Chapter 5 of Fielding's dissertation is [\[REST\]](#)).

RESTful web service is a web service implemented using HTTP and the principles of REST. It is a collection of resources, with four defined aspects:

- o The base URI for the web service, such as `http://example.com/resources/`.
- o The Internet media type of the data supported by the web service. This is often JSON, XML or YAML but can be any other valid Internet media type.
- o The set of operations supported by the web service using HTTP methods (e.g., GET, PUT, POST, or DELETE).
- o The API must be hypertext driven.

3. Response Format Specification

This section describes the response format specification for querying domain name, nameserver, contact and registrar. The specific format is defined in JSON. The basic querying URLs are described in the [draft-hollenbeck-dnrd-ap-query-01](#).

The following response objects defined MAY be supported optionally by each registry. Clients processing JSON responses MUST be prepared for values specified in this document to be absent from a response as no JSON value listed in this document is required to appear in the response. In other words, the server can remove the values as is needed by the policy of the registries.

3.1. Domain Name Response Specification

This section gives the set of required information and format that SHOULD be included in the domain name responses. An extension interface MAY also be defined for customized response information.

The corresponding querying URI example is:

`http://example.com/dnrd-ap/domain/example.com/` or
`http://example.com/dnrd-ap/example.com/`

The following is an elided example of a JSON response for a domain query:

```
{
  "domain": {
    ...
  },
  "nameserver": [
    ...
  ],
  "contact": [
    ...
  ],
  "registrar": {
    ...
  },
  "extension": {
    ...
  }
}
```

3.1.1. Objects in the Domain Name Response

There are "domain", "nameserver", "contact", "registrar" and "extension" objects included in the domain response. The followings are description and explanation of objects members.

domain -- contains the information about the domain. The members of domain object are defined as below.

- o domainName -- contains the fully qualified name of the registered domain.
- o roid -- the identifier assigned to the domain name when the object was created.
- o domainStatus -- an array contains the current status values defined in [[RFC5731](#)].
- o clientID -- element that contains the identifier of the sponsoring client.
- o createID -- contains the identifier of the client that created the domain.

- o `creationDate` -- the date and time when the domain name is registered. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o `expirationDate` -- the date and time when the domain name is expiry. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o `updateID` -- contains the identifier of the client that updated the domain.
- o `updateDate` -- the date and time when the domain name is updated. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o `transferDate` -- the date and time of the most recent successful transfer. The date and time values MUST be represented in Universal Coordinated Time (UTC). If no transfer occurs, this key MUST not be provided.
- o `dnssec` -- the status of DNSSEC deployment. A value of "Signed" or "Yes" means that DNSSEC has been deployed. A value of "Unsigned" or "No" means that DNSSEC has not been deployed yet.

`nameserver` -- an array that contains the information about the nameserver or host. If the array object is a nameserver, "nameserverName" and "nameserverUri" SHOULD be the members of the array object. If the array object is a host, "host" and "hostUri" SHOULD be the members of the array object. Nameserver information is specified before host information. The members of nameserver object are defined as below.

- o `nameserverName` -- the fully qualified names of the delegated host objects.
- o `nameserverUri` -- the access address for detailed information about nameserver.
- o `host` -- contains the fully qualified names of the subordinate host objects that exist under this superordinate domain object.
- o `hostUri` -- the access address for detailed information about hosts.

`contact` -- an array that contains the information about the contacts. There are 4 types of contacts, including "registrant", "admin", "tech" and "billing" JSON objects. The members of contact object are defined as below.

- o type -- the type of contact. They are "registrant", "admin", "tech" and "billing".
- o contactID -- the identifier of registrant, administrative contact, technical contact or billing contact.
- o contactUri -- the access address for detailed information about registrant, administrative contact, technical contact or billing contact.

registrar -- contains the information about the registrar. The members of registrar object are defined as below.

- o sponsoringRegistrar -- the name of the corresponding registrar.
- o registrarUri - the access address for detailed information about registrar.

3.1.2. Extension Interface

A "Extension" key SHOULD be included in the JSON format for customized extension information of each registry.

3.1.3. JSON Format

```
{
  "domain": {
    "domainName": "example.com",
    "roid": "20030310s10001s00012956",
    "domainStatus": [
      "serverDeleteProhibited",
      "serverUpdateProhibited",
      "serverTransferProhibited"
    ],
    "clientID": "ClientX",
    "createID": "ClientY",
    "creationDate": "2003-03-10T19:06:00Z",
    "expirationDate": "2018-03-10T19:06:00Z",
    "updateID": "ClientX",
    "updateDate": "2006-03-10T19:06:00Z",
    "transferDate": "",
    "dnssec": "Unsigned"
  },
  "nameserver": [
    {
      "nameserverName": "ns1.example.com",
```



```
    "nameserverUri":
    "http://example.com/dnrd-ap/nameserver/ns1.example.com/"
  },
  {
    "nameserverName": "ns2.example.com",
    "nameserverUri":
    "http://example.com/dnrd-ap/nameserver/ns2.example.com/"
  },
  {
    "nameserverName": "ns3.example.com",
    "nameserverUri":
    "http://example.com/dnrd-ap/nameserver/ns3.example.com/"
  },
  {
    "host": "ns1.example.com",
    "hostUri":
    "http://example.com/dnrd-ap/nameserver/ns1.example.com/"
  },
  {
    "host": "ns2.example.com",
    "hostUri":
    "http://example.com/dnrd-ap/nameserver/ns2.example.com/"
  },
  {
    "host": "ns3.example.com",
    "hostUri":
    "http://example.com/dnrd-ap/nameserver/ns3.example.com/"
  }
],
"contact": [
  {
    "type": "registrant",
    "contactID": "xyz",
    "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
  },
  {
    "type": "admin",
    "contactID": "xyz",
    "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
  },
  {
    "type": "tech",
    "contactID": "xyz",
    "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
  },
  {
    "type": "billing",
    "contactID": "xyz",
```



```
    "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
  }
],
"registrar": {
  "sponsoringRegistrar": "ABCRegistrar",
  "registrarUri": "http://example.com/dnrd-ap/registrar/ABCRegistrar/"
},
"extension": {
  ...
}
}
```

3.2. Nameserver Response Specification

This section gives the set of required information and format that SHOULD be included in the nameserver responses. An extension interface MAY also be defined for customized response information.

The corresponding querying URL is:

```
http://example.com/dnrd-ap/nameserver/ns1.example.com/
```

The following is an elided example of a JSON response for a nameserver query:

```
{
  "nameserver": {
    ...
  },
  "extension": {
    ...
  }
}
```

3.2.1. Objects in the Nameserver Response

nameserver -- contains the information about the domain. The members of nameserver object are defined as below.

- o roid -- the identifier assigned to the host object.
- o nameserverName -- contains the fully qualified names of the delegated host objects.

- o `nameserverStatus` -- the status of host object.
- o `ipAddress` -- contains the IPv4, IPv6, or both addresses.
- o `clientID` -- element that contains the identifier of the sponsoring client.
- o `createID` -- contains the identifier of the client that created the nameserver.
- o `creationDate` -- the date and time when the nameserver is registered. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o `updateID` -- contains the identifier of the client that updated the domain.
- o `updateDate` -- the date and time when the nameserver is updated. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o `transferDate` -- the date and time of the most recent successful transfer. The date and time values MUST be represented in Universal Coordinated Time (UTC). If no transfer occurs, this key MUST not be provided.

3.2.2. Extension Interface

A "Extension" key SHOULD be included in the JSON format for customized extension information of each registry.

3.2.3. JSON Format


```
{
  "nameserver": {
    "roid": "host397232",
    "nameserverName": "ns1.example.com",
    "nameserverStatus": [
      "serverDeleteProhibited",
      "serverUpdateProhibited"
    ],
    "ipAddress": [
      "192.0.2.123",
      "2001:0DB8::1"
    ],
    "clientID": "ClientX",
    "createID": "ClientY",
    "creationDate": "2006-09-27T09:27:00Z",
    "updateID": "ClientX",
    "updateDate": "2007-10-27T09:27:00Z",
    "transferDate": ""
  },
  "extension": {
    ...
  }
}
```

3.3. Contact Response Specification

This section gives the set of required information and format that SHOULD be included in the contact responses. An extension interface MAY also be defined for customized response information.

The corresponding querying URL is:

```
http://example.com/dnrd-ap/contact/CID-4005/
```

The following is an elided example of a JSON response for a contact query:

```
{
  "contact": {
    ...
  },
  "extension": {
    ...
  }
}
```


3.3.1. Objects in the Contact Response

contact -- contains the information about the contacts. The members of contact object are defined as below.

- o contactID -- contains the identifier of the contact.
- o roid -- contains the repository object identifier assigned to the contact.
- o contactStatus -- gives the status of the contact.
- o postalInfo -- contains the post and address information. Both internationalized and local addresses can be provided in this object. A "type" member is defined with the value of "int" to indicate internationalize address, and "loc" to indicate local address.
 - * type -- contains the type of address, "int" is internationalized address, "loc" is local address.
 - * contactName -- the name of the individual represented by the contact.
 - * organization -- the group or company that the contact works for.
 - * address -- contains the detailed address information.
 - + street -- the street of the contact's address.
 - + stateOrProvince -- the state or province of the contact's address.
 - + postalCode -- the postal code of the contact's address.
 - + countryCode -- the contact's contry code.
- o phoneNumber -- contains the contact's phone number.
- o faxNumber -- contains the contact's fax number.
- o email -- the email adress of the contact.
- o clientID -- element that contains the identifier of the sponsoring client.

- o createID -- contains the identifier of the client that created the contact.
- o creationDate -- the date and time when the contact is created. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o updateID -- contains the identifier of the client that updated the contact.
- o updateDate -- the date and time when the contact is updated. The date and time values MUST be represented in Universal Coordinated Time (UTC).
- o transferDate -- the date and time of the most recent successful transfer. The date and time values MUST be represented in Universal Coordinated Time (UTC). If no transfer occurs, this key MUST not be provided.

3.3.2. Extension Interface

A "Extension" key SHOULD be included in the JSON format for customized extension information of each registry.

3.3.3. JSON Format


```
{
  "contact": {
    "contactID": "CID-4005",
    "roid": "CID-4005-REP",
    "contactStatus": [
      "ok",
      "linked"
    ],
    "postalInfo": [
      {
        "type": "int",
        "contactName": "Smith",
        "organization": "ABC Corp.",
        "address": {
          "street": "South 4th Street",
          "stateOrProvince": "CN",
          "postalCode": "100000",
          "countryCode": "CN"
        }
      }
    ],
    "phoneNumber": "+001.8002250180",
    "faxNumber": "+001.8002250180",
    "email": "domainadmin@example.com",
    "clientID": "ClientX",
    "createID": "ClientY",
    "creationDate": "2006-09-27T09:27:00Z",
    "updateID": "ClientX",
    "updateDate": "2007-10-27T09:27:00Z",
    "transferDate": ""
  },
  "extension":{
    ...
  }
}
```

3.4. Registrar Response Specification

This section gives the set of required information and format that SHOULD be included in the contact responses. An extension interface MAY also be defined for customized response information.

The corresponding querying URL is:

<http://example.com/dnrd-ap/registrar/Example Registrar, Inc./>

The following is an elided example of a JSON response for a registrar

query:

```
{
  "registrar": {
    ...
  },
  "registrarContact": [
    ...
  ],
  "extension": {
    ...
  }
}
```

3.4.1. Objects in the Registrar Response

registrar -- contains the information about the registrar. The members of registrar object are defined as below.

- o registrarIANAID -- contains the registrar ID specified by IANA.
- o postalInfo -- contains the post and address information. Both internationalized and local addresses can be provided in this object. A "type" member is defined with the value of "int" to indicate internationalize address, and "loc" to indicate local address.
 - * type -- contains the type of address, "int" is internationalized address, "loc" is local address.
 - * registrarName -- the name of the group or company represented by the registrar.
 - * organization - the group or company that the contact works for.
 - * address -- contains the detailed address information.
 - + street -- the street of the registrar's address.
 - + stateOrProvince -- the state or province of the registrar's address.
 - + postalCode -- the postal code of the registrar's address.

- + countryCode -- the registrar's country code.
 - o phoneNumber -- contains the contact's phone registrar.
 - o faxNumber -- contains the contact's fax registrar.
 - o email -- the email address of the registrar.
 - o clientID -- element that contains the identifier of the sponsoring client.
 - o createID -- contains the identifier of the client that created the registrar.
 - o creationDate -- the date and time when the registrar is created. The date and time values MUST be represented in Universal Coordinated Time (UTC).
 - o updateID -- contains the identifier of the client that updated the registrar.
 - o updateDate -- the date and time when the registrar is updated. The date and time values MUST be represented in Universal Coordinated Time (UTC).
 - o transferDate -- the date and time of the most recent successful transfer. The date and time values MUST be represented in Universal Coordinated Time (UTC). If no transfer occurs, this key MUST not be provided.
- registrarContact -- an array contains the information of contacts associated with the registrar.
- o type -- the type of contact. They are "admin", "tech" and "billing".
 - o contactID -- the identifier of administrative contact, technical contact or billing contact.
 - o contactUri -- the access address for detailed information about administrative contact, technical contact or billing contact.

3.4.2. Extension Interface

A "Extension" key SHOULD be included in the JSON format for customized extension information of each registry.

[3.4.3.](#) **JSON Format**


```
{
  "registrar": {
    "registrarIANAID": "registrar1234",
    "postalInfo": [
      {
        "type": "int",
        "registrarName": "Example Registrar, Inc.",
        "organization": "Example Corp.",
        "address": {
          "street": "South 4th Street",
          "stateOrProvince": "CN",
          "postalCode": "100000",
          "countryCode": "CN"
        }
      }
    ],
    "phoneNumber": "+001.8002250180",
    "faxNumber": "+001.8002250180",
    "email": "domainadmin@example.com",
    "clientID": "ClientX",
    "createID": "ClientY",
    "creationDate": "2006-09-27T09:27:00Z",
    "updateID": "ClientX",
    "updateDate": "2007-10-27T09:27:00Z",
    "transferDate": ""
  },
  "registrarContact": [
    {
      "type": "admin",
      "contactID": "xyz",
      "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
    },
    {
      "type": "tech",
      "contactID": "xyz",
      "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
    },
    {
      "type": "billing",
      "contactID": "xyz",
      "contactUri": "http://example.com/dnrd-ap/contact/xyz/"
    }
  ],
  "extension":{
    ...
  }
}
```


4. Accept Header

Clients MAY use a MIME type, such as "application/json" to desire the JSON format of response. The server SHOULD respond the client with the proper MIME type in the accept header, and give the above response in JSON format.

5. Internationalization Considerations

5.1. Encoding Specification

The response format MUST has a key for defining encoding format, such as "encoding", to explicitly demonstrate the character encoding that the server can support. Both the U-label and A-label domain names can be included in the response.

5.2. IDN Variants

Internationalized Domain Names (IDNs), particularly variants, brought about the need for multiple domain names to be registered and managed as a single package, or registration bundle. The bundle variant domain names share some attributes, such as expiration date and the registrar. So when users update any properties within a bundle, all of the domains' attributes in the bundle will be changed.

The response SHOULD be able to be extended to support the IDN variants to show the list of the bundle variant domain names.

6. IANA Considerations

This document does not specify any IANA actions.

7. Security considerations

TBD

8. Acknowledgements

This document has been reviewed and improved by the design team. The authors especially thank the following individuals who gave their suggestions and contributions to this document: Andy Newton and Steve Sheng.

9. References

9.1. Normative References

- [ECMA-262] "ECMAScript Language Specification", June 2011, <<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>>.
- [REST] Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", June 2000, <http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", [RFC 2622](#), June 1999.
- [RFC3912] Daigle, L., "WHOIS Protocol Specification", [RFC 3912](#), September 2004.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", [RFC 4627](#), July 2006.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, [RFC 5731](#), August 2009.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", [RFC 5890](#), August 2010.

9.2. Informative References

- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), August 2009.
- [RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, [RFC 5732](#), August 2009.
- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, [RFC 5733](#), August 2009.

Authors' Addresses

Ning Kong
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3147
Email: nkong@cnnic.cn

Linlin Zhou
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 2677
Email: zhoulinlin@cnnic.cn

Jiagui Xie
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 2639
Email: xiejiagui@cnnic.cn

Sean Shen
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3038
Email: shenshuo@cnnic.cn

