Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 2, 2018

A. Newton
ARIN

September 29, 2017

A Description of RDAP JSON Messages Using JSON Content Rules
draft-newton-rdap-jcr-03

Abstract

This document describes the JSON responses in the Registration Data Access Protocol with the formal notation of JSON Content Rules.

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 https://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 April 2, 2018.

Copyright Notice

Copyright (c) 2017 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 (https://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.

Table of Contents

1. Introduction 2
2. Response 3
3. Object Classes 4
3.1. Entity Object Class 4
3.2. Nameserver Object Class 4
3.3. Domain Object Class 5
3.4. IP Network Object Class 7
3.5. Autnum Object Class 7
4. Search Results 7
5. Common Structures 8
5.1. RDAP Conformance 8
5.2. Links 8
5.3. Notices And Remarks 9
5.4. Language Identifier 9
5.5. Events 9
5.6. Status 9
5.7. Port 43 10
5.8. Public IDs 10
6. Complete JCR for RDAP 10
7. Normative References 16
Author's Address 17

1. Introduction

The JSON [RFC7159] responses of the Registration Data Access Protocol [RFC7483] are officially defined with English prose. Those definitions contain imprecise or ambiguous JSON structures and require lengthy, tedious examples in the attempt to offer clarification. The English prose can be difficult for non-native English readers, and the examples create their own confusion.

This document describes the JSON found in RDAP with JSON Content Rules [I-D.newton-json-content-rules] (JCR).

JCR overcomes some of the obstacles of describing JSON with English prose, reducing the tediousness of the prose and accompanying lengthy examples to understandable data structures. Additionally, JCR has mechanisms which can be used by software developers to create test harnesses and technology compatibility kits.

Though this document describes all of the JSON found in [RFC7483], it presents the structures in a different order. The rules defined here use the JCR mixin style of specification, where common structures are defined in group rules instead of separately, distinct objects.

2. Response

[RFC7483] describes ten distinct JSON response: five entity class response, an error response, a help response, and three search responses.

```
$entity_response =           @{root} { $response_mixin,
                                       $entity_mixin }

$nameserver_response =       @{root} { $response_mixin,
                                       $nameserver_mixin }

$domain_response =           @{root} { $response_mixin,
                                       $domain_mixin }

$network_response =          @{root} { $response_mixin,
                                       $network_mixin }

$autnum_response =           @{root} { $response_mixin,
                                       $autnum_mixin }

$error_response =            @{root} { $response_mixin,
                                       $error_mixin }

$help_response =             @{root} { $response_mixin }

$domainSearch_response =     @{root} { $response_mixin,
                                       $domainSearchResult }

$nameserverSearch_response = @{root} { $response_mixin,
                                       $nameserverSearchResult }

$entitySearch_response =     @{root} { $response_mixin,
                                       $entitySearchResult }
```

All of the responses have a common set of object members described by response_mixin.

```
$response_mixin = (
   $rdapConformance ?,
   "notices" : $notices ?,
   $lang
)
```

3. Object Classes

The primary data structures in RDAP are called object classes. These are first order object instances with identifiers. They are JSON objects which contain other JSON data types.

3.1. Entity Object Class

The Entity object class represents persons or organizations. It incorporates jCard [RFC7095] (vCard in JSON) for contact information. The rules supplied here only provide for a basic validation of jCard, as the validation of jCard is beyond the scope of this document.

```
$entities = "entities" : [ $entity_oc * ]

$entity_oc = {
   $entity_mixin
}

$entity_mixin = (
   "objectClassName" : "entity",
   $common_mixin,
   "vcardArray"      : [ "vcard", [ $vcard * ] ] ?,
   "asEventActor"    : [ $event * ] ?,
   "roles"           : [ string * ] ?,
   $publicIds ?,
   $entities ?,
   "networks"        : [ $network_oc * ] ?,
   "autnums"         : [ $autnum_oc * ] ?
)

$vcard = @{unordered} [
   [ "version", {}, "text", "4.0" ],
   [ "fn", {}, "text", string ],
   [ string,
     { /.*/:any * },
     "text",
     ( string | [ string * ] )
   ] *
]
```

3.2. Nameserver Object Class

The nameserver object class represents DNS nameservers in registries.

```
$nameservers = "nameservers" : [ $nameserver_oc * ]

$nameserver_oc = {
   $nameserver_mixin
}

$nameserver_mixin = (
   "objectClassName" : "nameserver",
   $common_mixin,
   "ldhName"         : fqdn,
   "unicodeName"     : idn ?,
   "ipAddresses"     : { "v4" : [ ipv4 ? ] ?, "ip6" : [ ipv6 ? ] ? } ?,
   $entities ?
)
```

3.3. Domain Object Class

The Domain object class is the most complex of all the object classes defined in RDAP. It represents both forward and reverse DNS delegations. It's complexity is mostly due to the DNSSEC provisions of the object class.

```
$domain_oc = {
   $domain_mixin
}

$domain_mixin = (
   "objectClassName" : "domain",
   $common_mixin,
   "ldhName"         : fqdn,
   "unicodeName"     : idn ?,
   "variants"        : [ $variant * ] ?,
   $nameservers ?,
   $secureDNS ?
)

$variant = {
   "relation"     : [ string * ] ?,
   "idnTable"     : string ?,
   "variantNames" : [ { "ldhName" : fqdn, "unicodeName" : idn } * ]
}

$secureDNS = "secureDNS" : {
   "zoneSigned"       : boolean ?,
   "delegationSigned" : boolean ?,
   "maxSigLife"       : integer ?,
   "dsData"           : [ $dsData_obj * ] ?,
   "keyData"          : [ $keyData_obj * ] ?,
   $entities ?,
   $publicIds ?,
   "network"          : $network_oc ?
}

$dsData_obj = {
   "keyTag"     : integer,
   "algorithm"  : integer,
   "digest"     : string,
   "digestType" : integer,
   $events ?,
   $links ?
}

$keyData_obj = {
   "flags"     : integer,
   "protocol"  : integer,
   "publicKey" : string,
   "algorithm" : integer,
   $events ?,
   $links ?
}
```

3.4. IP Network Object Class

The IP Network object class represents IP network registrations in RIRs.

```
$network_oc = {
   $network_mixin
}

$network_mixin = (
   "objectClassName" : "ip network",
   $common_mixin,
   "startAddress"    : ( ipv4 | ipv6 ) ?,
   "endAddres"       : ( ipv4 | ipv6 ) ?,
   "ipVersion"       : ( "v4" | "v6" ) ?,
   "name"            : string ?,
   "type"            : string ?,
   "country"         : /[A-Z]{2}/ ?,
   "parentHandle"    : string ?,
   $entities ?
)
```

3.5. Autnum Object Class

The Autnum object class represents an autonomous system number or blocks of autonomous system numbers in an RIR.

```
$autnum_oc = {
   $autnum_mixin
}

$autnum_mixin = (
   "objectClassName" : "autnum",
   $common_mixin,
   "startAutnum"     : int32 ?,
   "endAutnum"       : int32 ?,
   "name"            : string ?,
   "type"            : string ?,
   $entities ?
)
```

4. Search Results

Search results in RDAP are merely arrays of object classes.

```
$domainSearchResult     = "domainSearchResult"     : [ $domain_oc * ]
$nameserverSearchResult = "nameserverSearchResult" : [ $nameserver_oc * ]
$entitySearchResult     = "entitySearchResult"     : [ $entity_oc * ]
```

5. Common Structures

Section 4 of [RFC7483] describes eight common structures used throughout the JSON in RDAP.

Most of these common structures are grouped together in a rule called common_mixin.

```
$common_mixin = (
   "handle"  : string ?,
   "remarks" : [ $notice * ] ?,
   $links,
   $events ?,
   $status ?,
   $port43 ?,
   $lang
)
```

5.1. RDAP Conformance

The rdapConformance array is the versioning and capabilities negotiation mechanism of RDAP.

```
$rdapConformance = "rdapConformance" : [ string * ]
```

5.2. Links

Structures in RDAP may link to information in other data systems using links. Additionally, RDAP uses "self" links to identify instances of RDAP object classes. The data found in each link is described by [RFC5988].

RDAP links are an array of distinct objects, each representing a separate link.

```
$links = ( "links" : [ $link * ] ? )

$link = {
   "value"    : uri ?,
   "rel"      : string ?,
   "href"     : uri,
   "hreflang" : [ $lang_value * ] ?,
   "title"    : string ?,
   "media"    : string ?,
   "type"     : /[a-zA-Z][a-zA-Z0-9]*\/[a-zA-Z][a-zA-Z0-9]*/ ?
}
```

5.3. Notices And Remarks

In RDAP, notices and remarks share the same structure. The difference is that notices are meta-data regarding the entirety of a response whereas remarks are meta-data covering a specific instance of an object class.

```
$notices = [ $notice * ]

$notice = {
    "title"       : string ?,
    "description" : [ string * ],
    "type"        : string ?,
    $links,
    $lang
}
```

5.4. Language Identifier

The "lang" member occurs many RDAP data structures. And the same construct is used in the links structures.

```
$lang_value =: /[a-z]{2}(\-[A-Z][a-zA-Z]*(\-[A-Z]{2})?)?/
$lang = ( "lang" : $lang_value ? )
```

5.5. Events

RDAP events note when a specific action has occured on an object instance, and by whom. The same structure appears in all object classes, as well as being re-used by entities embedded by other objects.

```
$events = "events" : [ $event * ]

$event = {
   "eventAction" : string,
   "eventActor"  : string ?,
   "eventDate"   : datetime,
   $links,
   $lang
}
```

5.6. Status

The status of RDAP object instances is indicated by an array of strings, where the value of the strings are registered in an IANA registry.

```
$status = "status" : [ string * ]
```

5.7. Port 43

RDAP object classes reference their corresponding Whois representation using the "port43" object member. This is simply a string holding the hostname of the Whois service.

```
$port43 = "port43" : string
```

5.8. Public IDs

Some RDAP services are required to identify entities and domains by public identifiers, such as ICANN Registrar IDs. The publicIds object member is an array of objects to represent these identifiers.

```
$publicIds = "publicIds" : [ $publicId * ]

$publicId = {
  "type"       : string,
  "identifier" : string
}
```

6. Complete JCR for RDAP

The following is the complete ruleset of JSON Content Rules for RDAP.

```
;
; The various types of responses
;

$entity_response =           @{root} { $response_mixin,
                                       $entity_mixin }

$nameserver_response =       @{root} { $response_mixin,
                                       $nameserver_mixin }

$domain_response =           @{root} { $response_mixin,
                                       $domain_mixin }

$network_response =          @{root} { $response_mixin,
                                       $network_mixin }

$autnum_response =           @{root} { $response_mixin,
                                       $autnum_mixin }

$error_response =            @{root} { $response_mixin,
                                       $error_mixin }
```

```
$help_response =             @{root} { $response_mixin }

$domainSearch_response =     @{root} { $response_mixin,
                                       $domainSearchResult }

$nameserverSearch_response = @{root} { $response_mixin,
                                       $nameserverSearchResult }

$entitySearch_response =     @{root} { $response_mixin,
                                       $entitySearchResult }

$response_mixin = (
   $rdapConformance ?,
   "notices" : $notices ?,
   $lang
)

;
; RFC 7483 Section 4.1 - RDAP Conformance
;

$rdapConformance = "rdapConformance" : [ string * ]

;
; RFC 7483 Section 4.2 - Links
;

$links = ( "links" : [ $link * ] ? )

; see RFC 5988
$link = {
   "value"    : uri ?,
   "rel"      : string ?,
   "href"     : uri,
   "hreflang" : [ $lang_value * ] ?,
   "title"    : string ?,
   "media"    : string ?,
   "type"     : /[a-zA-Z][a-zA-Z0-9]*\/[a-zA-Z][a-zA-Z0-9]*/ ?
}

;
; RFC 7483 Section 4.3 - Notices
;

$notices = [ $notice * ]

$notice = {
    "title"       : string ?,
```

```
    "description" : [ string * ],
    "type"        : string ?,
    $links,
    $lang
}

;
; RFC 7483 Section 4.4 - Language Identifier
;

$lang_value =: /[a-z]{2}(\-[A-Z][a-zA-Z]*(\-[A-Z]{2})?)?/
$lang = ( "lang" : $lang_value ? )

;
; RFC 7483 Section 4.5 - Events
;

$events = "events" : [ $event * ]

$event = {
   "eventAction" : string,
   "eventActor"  : string ?,
   "eventDate"   : datetime,
   $links,
   $lang
}

;
; RFC 7483 Section 4.6 - Status
;

$status = "status" : [ string * ]

;
; RFC 7483 Section 4.7 - Port43
;

$port43 = "port43" : string

;
; RFC 7482 Section 4.8 - Public Ids
;

$publicIds = "publicIds" : [ $publicId * ]

$publicId = {
  "type"       : string,
  "identifier" : string
```

```
}

;
; Common Object Class Structures
;

$common_mixin = (
   "handle"  : string ?,
   "remarks" : [ $notice * ] ?,
   $links,
   $events ?,
   $status ?,
   $port43 ?,
   $lang
)

;
; RFC 7483 Section 5.1 - Entity Object Class
;

$entities = "entities" : [ $entity_oc * ]

$entity_oc = {
   $entity_mixin
}

$entity_mixin = (
   "objectClassName" : "entity",
   $common_mixin,
   "vcardArray"      : [ "vcard", [ $vcard * ] ] ?,
   "asEventActor"    : [ $event * ] ?,
   "roles"           : [ string * ] ?,
   $publicIds ?,
   $entities ?,
   "networks"        : [ $network_oc * ] ?,
   "autnums"         : [ $autnum_oc * ] ?
)

; See RFC 7095
$vcard = @{unordered} [
   [ "version", {}, "text", "4.0" ],
   [ "fn", {}, "text", string ],
   [ string,
     { /.*/:any * },
     "text",
     ( string | [ string * ] )
   ] *
]
```

```
;
; RFC 7483 Section 5.2 - Nameserver Object Class
;

$nameservers = "nameservers" : [ $nameserver_oc * ]

$nameserver_oc = {
   $nameserver_mixin
}

$nameserver_mixin = (
   "objectClassName" : "nameserver",
   $common_mixin,
   "ldhName"         : fqdn,
   "unicodeName"     : idn ?,
   "ipAddresses"     : { "v4" : [ ipv4 ? ] ?, "ip6" : [ ipv6 ? ] ? } ?,
   $entities ?
)

;
; RFC 7483 Section 5.3 - Domain Object Class
;

$domain_oc = {
   $domain_mixin
}

$domain_mixin = (
   "objectClassName" : "domain",
   $common_mixin,
   "ldhName"         : fqdn,
   "unicodeName"     : idn ?,
   "variants"        : [ $variant * ] ?,
   $nameservers ?,
   $secureDNS ?
)

$variant = {
   "relation"     : [ string * ] ?,
   "idnTable"     : string ?,
   "variantNames" : [ { "ldhName" : fqdn, "unicodeName" : idn } * ]
}

$secureDNS = "secureDNS" : {
   "zoneSigned"       : boolean ?,
   "delegationSigned" : boolean ?,
   "maxSigLife"       : integer ?,
   "dsData"           : [ $dsData_obj * ] ?,
```

```
   "keyData"          : [ $keyData_obj * ] ?,
   $entities ?,
   $publicIds ?,
   "network"          : $network_oc ?
}

$dsData_obj = {
   "keyTag"     : integer,
   "algorithm"  : integer,
   "digest"     : string,
   "digestType" : integer,
   $events ?,
   $links ?
}

$keyData_obj = {
   "flags"     : integer,
   "protocol"  : integer,
   "publicKey" : string,
   "algorithm" : integer,
   $events ?,
   $links ?
}

;
; RFC 7483 Section 5.4 - IP Network Object Class
;

$network_oc = {
   $network_mixin
}

$network_mixin = (
   "objectClassName" : "ip network",
   $common_mixin,
   "startAddress"    : ( ipv4 | ipv6 ) ?,
   "endAddres"       : ( ipv4 | ipv6 ) ?,
   "ipVersion"       : ( "v4" | "v6" ) ?,
   "name"            : string ?,
   "type"            : string ?,
   "country"         : /[A-Z]{2}/ ?,
   "parentHandle"    : string ?,
   $entities ?
)

;
; RFC 7483 Section 5.5 - Autnum Object Class
;
```

```
$autnum_oc = {
   $autnum_mixin
}

$autnum_mixin = (
   "objectClassName" : "autnum",
   $common_mixin,
   "startAutnum"     : int32 ?,
   "endAutnum"       : int32 ?,
   "name"            : string ?,
   "type"            : string ?,
   $entities ?
)

;
; RFC 7483 Section 6 - Error
;

$error_mixin = (
   "errorCode"   : integer,
   "title"       : string ?,
   "description" : [ string * ] ?
)

;
; RFC 7483 Section 8 - Search Results
;

$domainSearchResult     = "domainSearchResult"     : [ $domain_oc * ]
$nameserverSearchResult = "nameserverSearchResult" : [ $nameserver_oc * ]
$entitySearchResult     = "entitySearchResult"     : [ $entity_oc * ]
```

Complete JSON Content Rules for RDAP

7. Normative References

[I-D.newton-json-content-rules]
Newton, A. and P. Cordell, "A Language for Rules Describing JSON Content", draft-newton-json-content-rules-09 (work in progress), September 2017.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <https://www.rfc-editor.org/info/rfc5988>.

[RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <https://www.rfc-editor.org/info/rfc7095>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[RFC7483] Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", RFC 7483, DOI 10.17487/RFC7483, March 2015, <https://www.rfc-editor.org/info/rfc7483>.

Author's Address

Andrew Lee Newton
American Registry for Internet Numbers
PO Box 232290
Centerville, VA 20120
US

Email: andy@arin.net
URI: http://www.arin.net