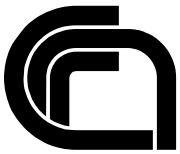

Consiglio Nazionale delle Ricerche

Extending RDAP Query Parameters for Result Sorting and Paging

M. Loffredo, M. Martinelli, S. Hollenbeck

IIT TR-05/2017

Technical Report

Giugno 2017

Istituto di Informatica e Telematica

Extending RDAP Query Parameters for Result Sorting and Paging

Loffredo M., Martinelli M. – IIT CNR
Hollenbeck S. – Verisign Labs

Summary

Abstract

The Registration Data Access Protocol (RDAP) does not include core functionality for clients to provide sorting and paging (subsetting) parameters for control of large result sets. This omission can lead to unpredictable server processing of queries and client processing of responses. This unpredictability can be greatly reduced if clients can provide servers with their preferences for managing response values. This document describes RDAP query extensions that allow clients to specify their preferences for sorting and paging result sets.

1. Introduction

The availability of functionality for result sorting and paging provides benefits to both clients and servers in the implementation of RESTful services [REST]. These benefits include:

- Reducing the server response bandwidth requirements;
- Improving server response time;
- Improvement in query precision and, consequently, obtaining more reliable results;
- Decreasing server query processing load.

Approaches to implementing features for result sorting and paging can be grouped into two main categories:

1. Sorting and paging are implemented through the introduction of additional parameters in the query string (i.e. ODATA protocol [OData-Part1]);
2. Information related to the number of results and the specific portion of the result set to be returned, in addition to a set of ready-made links for the result set scrolling, are inserted in the HTTP header of the request/response.

However, there are some drawbacks associated with use of the HTTP header. First, the header properties cannot be set directly from a web browser. Moreover, in an HTTP session, the information on the status (i.e. the session identifier) is usually inserted in the header or in the cookies, while the information on the resource identification or the search type is included in the query string. The second approach is therefore not compliant with the HTTP standard [RFC 2616]. As a result, this document describes a specification based on use of query parameters.

Currently the RDAP protocol [RFC7482] defines two query types:

- lookup: the server returns only one object;
- search: the server returns a collection of objects.

While the lookup query does not produce issues in the management of large result sets, the search query can potentially generate a large result set that could be truncated according to the limits of the server. In addition, it is not possible to obtain the total number of the objects found that might be returned in a search query response [RFC 7483]. Lastly, there is no mechanism to specify sort criteria to return the most relevant objects at the beginning of the result set.

The protocol described in this specification extends RDAP query capabilities to enable result sorting and paging, by adding four new path segments (i.e. search paths) and using a RESTful web service. The service is implemented using the Hypertext Transfer Protocol (HTTP) [RFC7230] and the conventions described in RFC 7480 [RFC7480].

The implementation of these parameters is technically feasible, as operators for counting, sorting and paging rows are currently supported by the major RDBMSs. Impact on the current state of RDAP implementation is estimated to be quite low.

2. RDAP Path Segment Specification

The new path segments are OPTIONAL extensions of path segments defined in RFC 7482 [RFC7482]. The path segments are:

- "count": which allows to obtain, as an additional information in the response, the number of objects found (that due to truncation can be different from the number of returned objects);
- "sortby": which allows to specify the sort criteria for the result set;
- "limit" and "offset": which allow to specify what portion of the entire result set is requested and to use the "links" property to provide a ready-made reference to the next page of the result set.

Augmented Backus–Naur Form (ABNF) [RFC5234] is used in the following sections to describe the formal syntax of these new parameters.

2.1 "count" parameter

Currently the RDAP protocol does not allow a client to determine the total number of the results in a query response when the result set is truncated. This is rather inefficient because the user can't evaluate the query precision and, at the same time, can't receive information that could be relevant.

The new parameter "count" provides additional functionality (Figure 1) that allows a client to request information from the server that specifies the number of elements found to match a particular search pattern.

https://rdap.pubtest.nic.it/domains?name=*nr.it&count=true

Figure 1. Example of RDAP query reporting the count parameter

The ABNF syntax is the following:

```
count = "count" EQ ( trueValue / falseValue )
trueValue = ("true" / "yes" / "1")
falseValue = ("false" / "no" / "0")
EQ = "="
```

A *trueValue* means that the server MUST provide the total number of the objects in the `paging_count` property of the response (Figure 2). A *falseValue* means that the server MUST NOT provide this number.

```
{
      "rdapConformance": [
            "rdap_level_0",
            "paging_level_0"
      ],
      ...
      "paging_count":"73",
      "domainSearchResults": [
            ...
  ]
}
```

Figure 2. Example of RDAP response with paging_count property

2.2 "sortby" parameter

The RDAP protocol does not provide any capability to specify response value sort criteria. A server could implement a default sorting scheme according to the object class, but this feature is not mandatory and might not meet user requirements. Sorting can be addressed by the client, but this solution is rather inefficient. Sorting and paging using features provided by the DBMS used by the RDAP server could help avoid truncation of relevant results and allow for scrolling the result set using subsequent queries.

The "sortby" parameter allows the client to ask the server to sort the results according to the values of one or more properties and according to the sort direction of each property. The ABNF syntax is the following:

```
sortby     = "sortby" EQ sortbyItem *( "," sortbyItem )
sortbyItem = property-ref [":" ( "a" / "d" ) ]
```

'a' means that the ascending sort MUST be applied, 'd' means that the descending sort MUST be applied. If the sort direction is absent, an ascending sort MUST be applied (Figure 3).

In the sortby ABNF syntax, property-ref represents a reference to a property of an RDAP object. Such a reference could be expressed by using a JSON Path. The JSON Path in a JSON document [RFC7159] is equivalent to the XPath [W3C.CR-XPATH-31-20161213] in a XML document. For example, the JSON Path to select the value of the ldhName property inside an RDAP domain object is "$.ldhName**",** where $ identifies the root of the document (DOM). Another way to select a value inside a JSON document is the JSON Pointer [RFC6901]. While JSON Path or JSON Pointer are both standard ways to select any value inside JSON data, neither is particularly easy to use (e.g. "$.events[?(@.eventAction='registration')].eventDate" is the JSON Path expression of the registration date in a RDAP domain object).

Therefore, it is better to provide a definition of property-ref in terms of RDAP properties. However, not all the RDAP properties are suitable to be used in sort criteria, such as:

- properties providing service information (e.g. links, notices, remarks, etc.);
- multivalued properties (e.g. status, roles, variants, etc.);
- properties modelling relationships to other objects (e.g. entities);

On the contrary, some properties expressed as values of other properties (e.g. registration date) could be used in such a context.

In the following, the list of the most suitable properties for sort criteria is presented. The properties are divided in two groups, *objects common properties* and *objects specific properties*.

- Objects common properties
  Object common properties are derived from the merge of the "eventAction" and the "eventDate" property. The following values of sortby are defined:

- registrationDate
- reregistrationDate
- lastChangedDate
- expirationDate
- deletionDate
- reinstantiationDate
- transferDate
- lockedDate
- unlockedDate

- Objects specific properties
  With regard to the specific properties, some of them are already defined among the query paths. In the following the list of the sorting properties, grouped by objects, is shown:
  - Domain: ldhName.
  - Nameserver: ldhName, ipV4. ipV6.
  - Entity: fn, handle, org, email, tel, country, countryName, locality.

In the following, the correspondence between the values of sortby parameter and the RDAP object properties is shown (Table 1):

Object class	**sortby value**	**Object property**	**Reference in RFC 7483**	**Reference in RFC 6350**
Searchable objects	Common properties	eventAction values suffixed by "Date"	4.5	
Domain	ldhName	ldhName	5.3	
Nameserver	ldhName	ldhName	5.2	
	ipV4	v4 ipAddress	5.2	
	ipV6	v6 ipAddress	5.2	
Entity	Handle	handle	5.1	
	Fn	vcard fn	5.1	6.2.1
	org	vcard org	5.1	6.6.4
	tel	vcard tel with type="voice"	5.1	6.4.1
	email	vcard email	5.1	6.4.2
	country	country code (as given in ISO.3166.1988 [ISO.3166.1988]) of the country name in vcard adr	5.1	6.3.1
	countryName	country name in vcard adr	5.1	6.3.1
	locality	locality in vcard adr	5.1	6.3.1

Table 1: Sorting properties definition

With regard to the definitions in Table 1, some further considerations must be made to disambiguate cases where the RDAP object property is multivalued:

- Even if a nameserver can have multiple IPv4 and IPv6 addresses, the most common configuration includes one address for each IP version. Therefore, the assumption of having a single IPv4 and/or IPv6 value for a nameserver cannot be considered too stringent.
- With the exception of handle values, all the sorting properties defined for entity objects can be multivalued according to the definition of vCard as given in RFC6350 [RFC6350]. When more than a value is reported, sorting can be applied to the preferred value identified by the parameter pref="1".

Each RDAP provider MAY define other sorting properties than those shown in this document.

https://rdap.pubtest.nic.it/domains?name=*nr.it&sortby=ldhName

https://rdap.pubtest.nic.it/domains?name=*nr.it&sortby=registrationDate:d

https://rdap.pubtest.nic.it/domains?name=*nr.it&sortby=expirationDate,ldhName

Figure 3. Examples of RDAP query reporting the sortby parameter

2.3 “limit” and “offset” parameters

An RDAP query could return a response with hundreds of object return values, especially when partial matching is used. For that reason, two parameters addressing result pagination are defined to make responses easier to handle:

- “limit”: means that the server MUST return the first N objects of the result set in the response;
- “offset”: means that the server MUST skip the first N objects and MUST return objects starting from position N+1.

The ABNF syntax of the parameters is the following:

```
limit = "limit" EQ positive-number
offset = "offset" EQ positive-number
positive-number = non-zero-digit *digit
non-zero-digit = "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
digit = "0" / non-zero-digit
```

When limit and offset are used together, they allow implementation of result pagination. These examples illustrate requests to return the first 5 objects, the set of objects starting from position 6, and 5 objects starting from position 11 of the result set (Figure 4).

https://rdap.pubtest.nic.it/domains?name=*nr.it&limit=5

https://rdap.pubtest.nic.it/domains?name=*nr.it&offset=5

https://rdap.pubtest.nic.it/domains?name=*nr.it&limit=5&offset=10

Figure 4. Examples of RDAP query reporting the limit and offset parameters

2.3.1. Use of "links" property

An RDAP server MAY use the “links” [RFC5988] property to provide a ready-made reference to the next page of the result set (Figure 5).

Examples of "rel" values are "first", "last", "prev".

```
{
      "rdapConformance": [
            "rdap_level_0",
            "paging_level_0"
      ],
      ...
      "notices": [
            {
            "title":"Search query limits",
            "type":"result set truncated due to excessive load",
            "description": [
                  "search results for domains are limited to 10"
            ]
            }
      ],
      "links": [
            {
            "value":"https://rdap.pubtest.nic.it/domains?name=*nr.it&count=true"
                  ,
            "rel":"next",
            "href":"https://rdap.pubtest.nic.it/domains?name=*nr.it&count=true
                  &limit=10&offset=10",
            "title": [
                 "Result Pagination Link"
            ]
            "type":"application/rdap+json"
            }
      ],
      "paging_count":"73",
      "domainSearchResults": [
            ...
  ]
}
```

Figure 5. Example of “links” property to implement result pagination

3. Negative answers

The value constraints for the parameters are defined by their ABNF syntax. Therefore, each request providing an invalid value for a parameter SHOULD obtain an HTTP 400 (Bad Request) response code. The same response SHOULD be returned if the client provides an unsupported value for the sortby parameter.

The server can provide a different response when it supports the limit and/or offset parameters and the client submits values that are out of the valid ranges. The possible cases are:

- If the client submits a value for the limit parameter that is greater than the number of objects to be processed, it is RECOMMENDED that server returns a response including only the processed objects.
- If the client submits a value for the offset parameter that is greater than the number of objects to be processed, it is RECOMMENDED that server returns an HTTP 404 (Not Found) response code.

Optionally, the response MAY include additional information regarding the negative answer in the HTTP entity body.

4. RDAP conformance

Servers implementing the count parameter MUST include "paging_level_0" in the *rdapConformance* array of their responses, when the count parameter is set to a *trueValue*.

5. Implementation Considerations

The implementation of the new parameters is technically feasible, as operators for counting, sorting and paging are currently supported by the major RDBMS.

In the following, the match between the new defined parameters and the SQL operators is shown (Table 2):

New query parameter	SQL operator
Count	count(*) query without offset, limit and order by [MYSQL-COUNT], [POSTGRES-COUNT], [ORACLE-COUNT]
Sortby	order by [MYSQL-SORT], [POSTGRES-SORT], [ORACLE-SORT]
Limit	limit n (in MySql [MYSQL-LIMIT] and Postgres [POSTGRES-LIMIT]) `FIRST n ROWS ONLY` (in Oracle [ORACLE-LIMIT])
Offset	offset m (in Postgres) OFFSET m ROWS (in Oracle)
limit + offset	limit n offset m (in MySql and Postgres) `OFFSET m ROWS FETCH NEXT n ROWS ONLY` (in Oracle)

Table 2: New query parameters vs. SQL operators

With regards to the Oracle, the Table 2 reports only one of the three methods that can be used to implement offset and limit. The others are described in [ORACLE-ROWNUM] and [ORACLE-ROW-NUMBER].

In addition, similar operators are completely or partially supported by the most known NoSQL databases (MongoDB, CouchDB, HBase, Cassandra, Hadoop) so the implementation of the new parameters seems to be practicable by servers working without the use of an RDBMS.

6. IANA Considerations

An IANA-maintained registry of all the properties that could be used in sort criteria is worth discussing. Such a registry could be also adopted to identify the properties to be used in future RDAP query filtering capabilities. The contents of this registry is described in 5.1 Registry of references to RDAP properties.

6.1 Registry of the references to RDAP properties

Entries in this registry could contain the following:

Object class name: the class name of the object the property belongs to (as defined in in [RFC7483]);

Property reference: the reference to the property;

JSON Path: the JSON path of the property inside the object (as defined in [RFC7159]);

Published specification: RFC number, bibliographical reference, or URL to a permanent and readily available specification;

Contact: The names and email addresses of individuals to contact regarding this registry entry;

Intended usage: brief reasons for this registry entry (as defined by [RFC5226]).

In the following, the example of the entry related to the reference “ipV4” is reported:

Object class name: Nameserver

Property reference: ipV4

JSON Path: $.ipAddresses.v4[0]

Published specification: https://www.ietf.org/id/draft-loffredo-regext-rdap-sorting-and-paging-00.txt

Contact: mario.loffredo@iit.cnr.it

Intended usage: The entry is about the reference to the v4 address within a nameserver object.

7. Security Considerations

Security services for the operations specified in this document are described in [RFC7481].

Search query typically requires more server resources (such as memory, CPU cycles, and network bandwidth) when compared to lookup query. This increases the risk of server resource exhaustion and subsequent denial of service due to abuse. This risk can be mitigated by either restricting search functionality and limiting the rate of search requests. Servers can also reduce their load by truncating the results in the response. However, this last security policy can result in a higher inefficiency if the RDAP server does not provide any functionality to return the truncated results.

The new parameters presented in this document provide the RDAP operators with a way to implement a secure server without penalizing its efficiency. The “count” parameter gives the user a measure to evaluate the query precision and, at the same time, return a significant information. The “sortby” parameter allows the user to obtain the most relevant information at the beginning of the result set. In both cases, the user doesn’t need to submit further unnecessary search requests. Finally, the “limit” and “offset” parameters enable the user to scroll the result set by submitting a sequence of sustainable queries according to the server limits.

8. References

8.1 Normative References

[ISO.3166.1988] International Organization for Standardization, "Codes for the representation of names of countries, 3rd edition", ISO Standard 3166, August 1988.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee," Hypertext Transfer Protocol -- HTTP/1.1", June 1999, <https://tools.ietf.org/html/rfc2616>

[RFC5234] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF", January 2008, <https://tools.ietf.org/html/rfc5234>

[RFC5266] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs ", May 2008, https://tools.ietf.org/html/rfc5266

[RFC5988] M. Nottingham,"Web Linking", DOI 10.17487/RFC7233, October 2010, <https://tools.ietf.org/html/rfc5988>

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <http://www.rfc-editor.org/info/rfc6350>

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

[RFC7230] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>

[RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", RFC 7480, March 2015, <http://www.rfc-editor.org/info/rfc7480>.

[RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", RFC 7481, DOI 10.17487/RFC7481, March 2015, <http://www.rfc-editor.org/info/rfc7481>.

[RFC7482] A. Newton and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", RFC 7482, DOI 10.17487/RFC7482, March 2015, <http://www.rfc-editor.org/info/rfc7482>.

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

8.2 Informative References

[MYSQL-COUNT] mysql.com, "MySQL 5.7 Reference Manual, Counting Rows", <http://dev.mysql.com/doc/refman/5.7/en/counting-rows.html>.

[MYSQL-LIMIT] mysql.com, "MySQL 5.7 Reference Manual, SELECT Syntax", <https://dev.mysql.com/doc/refman/5.7/en/select.html>.

[MYSQL-SORT] mysql.com, "MySQL 5.7 Reference Manual, MySQL Sorting Rows", <https://dev.mysql.com/doc/refman/5.7/en/sorting-rows.html>.

[ODATA-PART1] OData Version 4.0. Part 1: Protocol Plus Errata 03. Edited by Michael Pizzo, Ralf Handl, and Martin Zurmuehl. 02 June 2016. OASIS Standard incorporating Approved Errata 03. http://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part1-protocol/odata-v4.0-errata03-os-part1-protocol-complete.html. Latest version: http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part1-protocol.html.

[ORACLE-COUNT] Oracle Corporation, "Database SQL Reference, COUNT", <http://docs.oracle.com/database/122/SQLRF/COUNT.htm>.

[ORACLE-LIMIT] Oracle Corporation, "Database SQL Language Reference, SELECT, Row limiting clause", <http://docs.oracle.com/database/122/SQLRF/SELECT.htm>.

[ORACLE-ROWNUM] Oracle Corporation, "Database SQL Language Reference, SELECT, ROWNUM Pseudocolumn", <http://docs.oracle.com/database/122/SQLRF/ROWNUM-Pseudocolumn.htm#SQLRF00255>.

[ORACLE-ROW-NUMBER] Oracle Corporation, "Database SQL Language Reference, ROW_NUMBER", <http://docs.oracle.com/database/122/SQLRF/ROW_NUMBER.htm#SQLRF06100>.

[ORACLE-SORT] Oracle Corporation, "Database SQL Reference, SELECT, order by clause", <http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_10002.htm>.

[POSTGRES-COUNT] postgresql.org, "PostgreSQL, Aggregate Functions", <https://www.postgresql.org/docs/9.5/static/functions-aggregate.html>.

[POSTGRES-LIMIT] postgresql.org, "PostgreSQL, LIMIT and OFFSET", <https://www.postgresql.org/docs/9.5/static/queries-limit.html>.

[POSTGRES-SORT] postgresql.org, "PostgreSQL, Sorting Rows", <https://www.postgresql.org/docs/9.5/static/queries-order.html>.

[REST] T. FREDRICH, "RESTFUL SERVICE BEST PRACTICES, RECOMMENDATIONS FOR CREATING WEB SERVICES”, <HTTP://WWW.RESTAPITUTORIAL.COM/MEDIA/RESTFUL_BEST_PRACTICES-V1_1.PDF>

[RFC6901] P. BRYAN, K. ZYP, M. NOTTINGHAM, "JAVASCRIPT OBJECT NOTATION (JSON) POINTER", DOI 10.17487/RFC6901, APRIL 2013, <HTTPS://WWW.RFC-EDITOR.ORG/INFO/RFC6901>.

[W3C.CR-XPATH-31-20161213] "XML PATH LANGUAGE (XPATH) 3.1", DECEMBER 2016, <HTTPS://WWW.W3.ORG/TR/2016/CR-XPATH-31-20161213>