

# Extending RDAP query parameters to enable result sorting and paging

Mario Loffredo, Maurizio Martinelli  
IIT-CNR/Registro.it

- REST services
- RDAP status
- New query parameters
- Considerations
  - Implementation
  - Security
- Q & A

- RESTful services SHOULD offer capabilities for result filtering, sorting and paging in order to:
  - Minimize the traffic (requests/responses) on the net
  - Speed up the response time
  - Improve the precision of the queries and, consequently, obtain more reliable results
  - Decrease the load of the server in the query processing

## ■ Current approaches

- Sorting and paging are implemented through the introduction of additional parameters in the query string (i.e. ODATA protocol)
  - 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
- 
- The use of HTTP header is not recommended because:
    - properties cannot be set directly from a web browser
    - in a HTTP session,
      - information on the status (e.g. session ID) are inserted in the header or in the cookies
      - information on the resource identification or the search type are included in the query string

- The search query can return a large result set
- The result set can be truncated according to the limits of the server
- RDAP lacks of result filtering, sorting and paging capabilities:
  - you cannot restrict the result set by adding further search conditions
  - you cannot obtain, in the response, the total number of the objects found in order to evaluate the query precision
  - you cannot specify possible sort criteria:
    - to have the most relevant objects at the beginning of the result set
    - to avoid the truncation of relevant results
  - you cannot scroll the result set by subsequent queries when result set is truncated

- New “Boolean search capabilities” have been announced to address filtering
- Registro.it implementation defines four new parameters to enable result sorting and paging:
  - **count**: 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**: allows to specify a sort order for the result set
  - **limit & offset**: allow to specify what portion of the entire result set must be returned and to use the “links” property to provide a ready-made reference to the next page of the result set

## ■ ABNF syntax

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

- The server provides the total number of the objects in the *paging\_count* property of the response

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

## ■ Samples

- [https://rdap.pubtest.nic.it/entities?handle=R\\*-REG&count=true](https://rdap.pubtest.nic.it/entities?handle=R*-REG&count=true)
- [https://rdap.pubtest.nic.it/domains?name=we\\*.it&count=1](https://rdap.pubtest.nic.it/domains?name=we*.it&count=1)

## ■ ABNF syntax

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

- a means ascending sort (default)
- d means descending sort

## ■ How to define property-ref ?

- By JSON Path/Pointer ?
  - Not easy to use (e.g. "\$.events[?(@.eventAction='registration')].eventDate")
- It is better to define it in terms of RDAP properties
  - Some properties are not suitable to be used in sort criteria
    - properties (e.g. links, notices, remarks, etc.) providing service information
    - multivalued properties (e.g. status, roles, variants, etc.)
    - properties (e.g. entities) modelling relationships to other objects
  - Some suitable properties are expressed as values of other properties (e.g. registration date)
  - If exist, EPP properties can be used as aliases
    - EPP properties are known better than the equivalent RDAP properties
    - EPP properties are generally shorter than the equivalent RDAP properties



## ■ Objects common properties:

- derived by values of the “eventAction” property in the “events” array
  - registrationDate (crDate)
  - lastChangedDate (upDate)
  - expirationDate (exDate)
  - transferDate (trDate)
  - deletionDate
  - ...

## ■ Objects specific properties:

- some of them are already present among the query paths
  - **Domain:** ldhName, nsLdhName, nslp
  - **Nameserver:** ldhName, ip
  - **Entity:** fn, handle, org, email, tel (voice), country (cc), countryName, locality (city)
  - **Autnum:** handle, startAutnum, type, name, country (cc)
  - **Ip:** handle, startAddress, ipVersion, type, name, country (cc)

- RDAP providers may define additional sorting properties
  
- A IANA-maintained registry of all the properties that could be used in sort criteria is recommended
  - Some of the properties in the registry could be the same used in future RDAP query filtering capabilities
  
- Samples
  - [https://rdap.pubtest.nic.it/domains?name=we\\*.it&orderby=crDate:d](https://rdap.pubtest.nic.it/domains?name=we*.it&orderby=crDate:d)
  - [https://rdap.pubtest.nic.it/domains?name=we\\*.it&orderby=ldhName](https://rdap.pubtest.nic.it/domains?name=we*.it&orderby=ldhName)
  - [https://rdap.pubtest.nic.it/entities?handle=R\\*-REG&orderby=org](https://rdap.pubtest.nic.it/entities?handle=R*-REG&orderby=org)
  - [https://rdap.pubtest.nic.it/entities?handle=R\\*-REG&orderby=handle](https://rdap.pubtest.nic.it/entities?handle=R*-REG&orderby=handle)
  - [https://rdap.pubtest.nic.it/entities?handle=R\\*-REG&orderby=cc,city](https://rdap.pubtest.nic.it/entities?handle=R*-REG&orderby=cc,city)

## ■ ABNF syntax

```
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
```

- By limit, the server returns the first N objects of the result set
- By offset, the server returns the next objects starting from N+1-th

## ■ The server can use the “links” property to provide a ready-made reference to the next page when results are truncated

```
"links": [
  {
    "value": "https://rdap.pubtest.nic.it/domains?name=we*.it",
    "rel": "next",
    "href":
"https://rdap.pubtest.nic.it/domains?name=we*.it&limit=10&offset=10",
    "type": "application/rdap+json"
  }
]
```

## ■ Samples

- [https://rdap.pubtest.nic.it/domains?name=we\\*.it&count=true](https://rdap.pubtest.nic.it/domains?name=we*.it&count=true)

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

RDAP query parameter	SQL operator
limit	limit n (MySQL and Postgres) ROW_NUMBER()<=n (Oracle)
offset	offset m (Postgres) ROW_NUMBER()>m (Oracle)
limit + offset	limit n offset m (MySQL and Postgres) ROW_NUMBER()>m AND ROW_NUMBER()<m+n (Oracle)
sortby	order by
count	“select count(*)“ query without offset, limit and order by

- RDAP conformance

- Servers implementing the count parameter must include “paging\_level\_0” in the rdapConformance array of their responses when the count property is set to a “true” value

- Search query typically requires more server resources than lookup query
- This increases the risk of server resource exhaustion and subsequent denial of service due to abuse
- Risks can be mitigated by:
  - restricting search functionality
  - limiting the rate of search requests
  - truncating the results in the response
- Truncation can result in a higher inefficiency if servers are not able to:
  - return the truncated results
  - provide the most relevant results at the beginning of the result set

- The new query capabilities support security without reducing efficiency
  - **count** gives a measure to evaluate the query precision and returns a relevant information
  - **orderby** helps to obtain most relevant information at the beginning of the result set
  - in both above cases, the user doesn't need to submit further unnecessary search requests
  - **limit** and **offset** allow to scroll the result set by submitting a sequence of sustainable queries according to the server limits

Thanks for your attention!  
Q & A