6th ROW – Registration Operations Workshop
May 12th, 2017
Madrid, Spain

Extending RDAP query parameters

to enable result sorting and paging

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

Abstract

The availability of parameters for result sorting and paging is considered one of the best practices in the implementation of RESTful services. The main reasons of the presence of these parameters are:

- 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.

Currently the different approaches to the implementation of what stated above, can be grouped in two main categories:

1. Sorting and paging are implemented through the introduction of additional parameters in the query string (i.e. ODATA protocol);
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 in the use of the HTTP header that recommend the adoption of the first approach. First of all, the header properties cannot be set directly from a web browser. Moreover, in a HTTP session, generally the information on the status (i.e. the session identifier) are inserted in the header or in the cookies, while the information on the resource identification or the search type are included in the query string. This approach is therefore not compliant with the HTTP standard (RFC 2616).

Currently the RDAP protocol defines two query types (RFC 7482):

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

While the *lookup query* does not arise particular issues in the management of the results, the *search query* can potentially generate a large result set that could be truncated according to the limits of the server. In addition, through a search query, it is not possible to obtain the total number of the objects found (RFC 7483). Lastly, it doesn't exist a mechanism to specify possible sort criteria, thus obtaining the most relevant objects at the beginning of the result set. A server could implement a default sorting according to the object class, but this feature is not mandatory and may not even meet the user requirements. Otherwise the sorting could be addressed by the client, but this solution is rather inefficient. In our opinion, the best

solution is therefore to implement sorting and paging on the DBMS used by the RDAP server. In this way, it could be possible to avoid the truncation of relevant results and scroll the result set by subsequent queries.

As an example of a RDAP server addressing the issues above, the Registro.it implementation (not yet publicly accessible) defines four new parameters for the search query:

- **sortby**: which allows to specify a sort order for the result set;
- **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);
- **offset** and **limit**: which 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.

The implementation of these new parameters is technically feasible, as operators for counting, sorting and paging rows are currently supported by the major RDBMS. Also the impact on the current state of the RDAP standard is quite low. The use of the count parameter introduces a basic type property in the response, while some properties of domain, nameserver and entity objects (which are the objects currently used by Registro.it), most likely to be used in sort criteria, have been identified and formalized as the sortby parameter values.