International

Virtual

Observatory

Alliance

Registry searching interface – REST API Version 1.0

IVOA Note 2013 May 08

Author(s):

Jonathan Normand
Pierre Le Sidaner

Abstract

Status of This Document

This is a Note. The is the first release of this document.

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents/.

Contents

1Introduction.....2
2Searching interface.....2
2.1URL scheme.....3
2.2Mandatory operations.....3
2.2.1GET /voresources/search.....3
2.2.1.1Parameters.....3
2.2.1.1.1keywords.....3
2.2.1.1.2orValues.....4
2.2.1.1.3IdentifiersOnly.....4
2.2.1.1.4From and max.....4
2.2.1.2Representation.....5
2.2.2GET /voresources.....5
2.2.2.1Representation.....6
2.2.3GET /identity.....6
2.3Error handling.....6
3To be discuss.....7
3.1Optional operations.....7
3.2Other relevant fields for search operation.....7
3.2.1provenance.....7
3.2.2updated.....7
3.3JSON representation.....7
3.3.1Content negotiation.....8
3.4Space coverage.....9
Appendix A.1 – Search XML response.....9
Appendix A.2 – Search XML response with identifiersOnly.....11

1 Introduction

The registry interface is composed of two independent parts. The harvesting interface provides the mechanism for registries to talk to each other and share information. The searching interface is used by clients that want to discover resources to use as part of a VO application. A registry may implement either interface or both, depending on the roles it intends to play.

This document focuses only on the searching interface and describes a REST API for it.

2 Searching interface

This interface uses an URL scheme using REST concepts and delivers XML-formatted responses via standard HTTP protocol. The operations that return resource descriptions do so using the RegistryInterface XML schema (http://www.ivoa.net/xml/RegistryInterface/v1.0).

2.1 URL scheme

```
{registry_endpoint}/{api_version}/{operation}
```

To take account of the next version of this search interface, the URL scheme must handle versioning. The version number of the API must be specify in the URL. For example:

http://example.com/registry/1/voresources/search

2.2 Mandatory operations

2.2.1 GET /voresources/search

Retrieves a list of resource descriptions matching a specified query. Only active resources are returned.

The search operation is applied to relevant fields of the resource that include at minimum: **identifier**, **description**, **title**, **type**, **subjects**, **publisher**, **shortname**, **waveband** and **standardid**. You can set constraints on specified fields or on all fields (cf. 2.2.1.1.1). By default when multiple constraints are set, the results of all constraints are combined with a logical AND.

As more than 13000 resources are registered it is necessary to handle pagination to give safely access to all resources (cf. 2.2.1.1.4).

URL structure	http://example.com/registry/1/voresources/search
Method	GET
Parameters	**Keywords:** required URL-encoded string (cf. 2.2.1.1.1) **orValues:** To combine multiple constraints with a logical OR. **From:** The minimum position in the complete list of matching resources **max:** The maximum number of resources to return **identifiersOnly:** To return only resource identifiers.
Return codes	200 OK (list of matching resources) 204 No Content. (when no resource matches the query) 400 Bad Request 405 Method Not Allowed 500 Server error
Representation	ri:VOResources

2.2.1.1 Parameters

2.2.1.1.1 keywords

The **keywords** syntax is similar to Lucene search syntax. In the examples below the strings are not URL-encoded for readability.

To search on all relevant fields, the string consists of one or more words separated by whitespace character. Example:

```
keywords=infrared
```

returns all resources containing the word “infrared” in at minimum one of the relevant fields.

To search on specific fields, the string consists of one or more field:value pairs separated by whitespace. If value contains “:” character the value string must be quoted. Example:

```
keywords=standardid:”ivo://ivoa.net/std/ConeSearch”
```

returns all resources having a ConeSearch interface.

```
keywords=standardid:”ivo://ivoa.net/std/ConeSearch” waveband:infrared
```

returns all resources having a ConeSearch interface and infrared in their waveband field.

2.2.1.1.2 orValues

By default multiple constraints search (ie. one for each word or field:value pair) uses the logical AND. If you want a logical OR, you have to add the **orValues** parameters in your query. As it is a boolean flag you do not need to give a value for this parameter. Example:

```
keywords=waveband:infrared description:HST&orValues
```

returns all resources having “HST” in their description field or having “infrared” in their waveband field.

2.2.1.1.3 IdentifiersOnly

Search operation can also return only resource identifier by using the **identifiersOnly** parameter. As it is a boolean flag you do not need to give a value for this parameter.

2.2.1.1.4 From and max

from and **max** parameters provide a pagination mechanism. By default the search operation returns its default number of resources starting at the first position in the complete list of matching resources. It can be modify by using **from** and **max** parameters. Example:

```
keywords=standardid:”ivo://ivoa.net/std/ConeSearch”&max=10&from=50
```

returns 10 ConeSearch starting from number 50.

2.2.1.2 Representation

The search operation returns a `ri:VOResources` element that contains each of the matching resources. The content of the `ri:VOResources` element depends on the presence of the **identifiersOnly** parameter. If **identifiersOnly** is put in the query string, then the `ri:VOResources` element must contain a list of ri:identifier elements. If no, then the `ri:VOResources` element must contain a list of `ri:Resource` elements of type Resource from the VOResource XML schema (http://www.ivoa.net/xml/VOResource/v1.0).

List of ri:Resource: full example available in appendix (cf. A.1)

```
<ri:VOResources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
   xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                       http://www.ivoa.net/xml/RegistryInterface/v1.0"
   numberReturned="5" more="true" from="1">
   <ri:Resource>...</ri:Resource>
   <ri:Resource>...</ri:Resource>
</ri:VOResources>
```

List of ri:identifier: full example available in appendix (cf. A.2)

```
<ri:VOResources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
   xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                       http://www.ivoa.net/xml/RegistryInterface/v1.0"
   numberReturned="5" more="true" from="1">
   <ri:identifier>...</ri:identifier>
   <ri:identifier>...</ri:identifier>
</ri:VOResources>
```

2.2.2 GET /voresources

Retrieves the full description of the resource specified by the **identifier** parameter.

URL structure	`http://example.com/registry/1/voresources`
Method	GET
Parameter	**`identifier:`** **required** The URL-encoded VO identifier of the resource you search
Return codes	200 OK (resource description) 405 Method Not Allowed 404 Resource not found 500 Server error
Representation	`ri:Resource`

Example: `/voresources?identifier=ivo://vopdc/registry`

2.2.2.1 Representation

This operation returns a `ri:Resource` elements of type Resource from the VOResource XML schema (http://www.ivoa.net/xml/VOResource/v1.0).

```
<ri:Resource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
   xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                       http://www.ivoa.net/xml/RegistryInterface/v1.0"
   created="..." status="..." updated="...">
...
</ri:Resource>
```

2.2.3 GET /identity

Retrieves the full description of the registry itself.

This is equivalent to GET /voresources operation where `identifier` is the identifier of this registry.

URL structure	`http://example.com/registry/1/identity`
Method	GET
Return codes	200 OK (registry description) 405 Method Not Allowed 500 Server error
Representation	`ri:Resource`

2.3 Error handling

Errors are returned using standard HTTP error code syntax (4XX, 5XX). Any additional information is included in the body of the return call, TEXT-formatted.

3 To be discuss

3.1 Optional operations

Each registries may also provide optional operations. To do that a new endpoint must be added. To allow users to discover the optional operations a `/capability` endpoint could be added.

3.2 Other relevant fields for search operation

Several useful fields can be added for management or technical search.

3.2.1 provenance

A **`provenance`** field allows to get all resources registered in a given registry defined by its IVO identifier. Example:

```
keywords=provenance:ivo://vopdc/registry
```

returns all resources registered in the VOPDC registry.

3.2.2 updated

An **`updated`** field allows to get all resources updated since a given date. Example:

```
keywords=updated:[2013 TO *]
```

returns all resources updated since 2013-01-01

3.3 JSON representation

The mandatory format for the response is XML using RegistryInterface schema. Sometimes clients want only get a subset of resource description. This is not easy to do it using a fixed schema. As JSON is schema free it could be used to do that. Here is an example of JSON-formatted representation for a resource.

The full description of the resource must include at least all the relevant fields used for searching (cf.2.2.1).

Full description: `/voresources?identifier=ivo://vopdc/registry`

```
{
   "status": "active",
   "publisher": "VO-Paris Data Centre",
   "updated": "2013-02-28T00:00:00Z",
   "description": "This registry provides access to resources from VO-
Paris Data Center and partners.",
   "title": "VO-Paris Data Center Publishing Registry",
   "provenance": "ivo://vopdc/registry",
   "referenceurl": "http://vo.obspm.fr",
   "created": "2013-01-15T00:00:00Z",
   "subjects": [
      "Registry",
      "Virtual Observatory"
   ],
   "capabilities": [
      {
         "accessurl": "http://voparis-registry.obspm.fr/oai/",
         "standardid": "ivo://ivoa.net/std/Registry"
      }
   ],
   "contactname": "J. Normand",
   "shortname": "VOPDC Registry",
   "identifier": "ivo://vopdc/registry",
   "type": "Registry"
}
```

Subset: `/voresources?identifier=ivo://vopdc/registry&fields=description,title,identifier`

```
{
   "description": "This registry provides access to resources from VO-
Paris Data Center and partners.",
   "title": "VO-Paris Data Center Publishing Registry",
   "identifier": "ivo://vopdc/registry",
}
```

3.3.1 Content negotiation

If the registry supports more than one format, a content negotiation mechanism must be added to the API. The HTTP specification already defined a content negotiation mechanism. The client provides an Accept HTTP header to select the best representation for it.

```
Accept: application/json
```

3.4 *Space coverage*

Appendix A.1 – Search XML response

http://example.com/registry/1/voresources/search?keywords=vopdc&max=2

```
<?xml version="1.0" encoding="UTF-8"?>
<ri:VOResources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
    xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                        http://www.ivoa.net/xml/RegistryInterface/v1.0"
    numberReturned="2" more="true" from="1">
    <ri:Resource created="2010-02-10T00:00:00Z" status="active"
        updated="2013-02-28T00:00:00Z"
        xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
        xmlns:vg="http://www.ivoa.net/xml/VORegistry/v1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                         http://www.ivoa.net/xml/RegistryInterface/v1.0
                         http://www.ivoa.net/xml/VORegistry/v1.0
                         http://www.ivoa.net/xml/VORegistry/v1.0"
        xsi:type="vg:Authority">
        <validationLevel validatedBy="ivo://vopdc">3</validationLevel>
        <title>VO-Paris Data Center Naming Authority</title>
        <shortName>VOPDC Authority</shortName>
        <identifier>ivo://vopdc</identifier>
        <curation>
            <publisher ivo-id="ivo://vopdc">VO-Paris Data
Centre</publisher>
            <creator>
                <name>Observatoire de Paris - VOPDC</name>
            </creator>
            <contact>
                <name>VOPDC support team</name>
                <address>Observatoire de Paris VOPDC, bat. Perrault, 77
av. Denfert Rochereau, 75014 Paris, FRANCE</address>
                <email>vo.paris@obspm.fr</email>
                <telephone>+33140512089</telephone>
            </contact>
        </curation>
        <content>
            <subject>Authority</subject>
            <subject>Virtual Observatory</subject>
            <description>This naming authority is the root of the
naming of the resources from
                VO-Paris Data Center and partners</description>
            <referenceURL>http://vo.obspm.fr</referenceURL>
            <type>Organisation</type>
            <contentLevel>General</contentLevel>
        </content>
        <managingOrg ivo-id="ivo://vopdc">VO-Paris Data
Centre</managingOrg>
```

```
    </ri:Resource>
    <ri:Resource created="2013-01-15T00:00:00Z" status="active"
updated="2013-02-28T00:00:00Z"
        xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
        xmlns:vg="http://www.ivoa.net/xml/VORegistry/v1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
                         http://www.ivoa.net/xml/RegistryInterface/v1.0
                         http://www.ivoa.net/xml/VORegistry/v1.0
                         http://www.ivoa.net/xml/VORegistry/v1.0"
        xsi:type="vg:Registry">
        <validationLevel validatedBy="ivo://vopdc">3</validationLevel>
        <title>VO-Paris Data Center Publishing Registry</title>
        <shortName>VOPDC Registry</shortName>
        <identifier>ivo://vopdc/registry</identifier>
        <curation>
            <publisher ivo-id="ivo://vopdc">VO-Paris Data
Centre</publisher>
            <creator>
                <name>J. Normand</name>
            </creator>
            <contributor>VOPDC</contributor>
            <contact>
                <name>VOPDC support team</name>
                <address>Observatoire de Paris VOPDC, bat. Perrault, 77
av. Denfert Rochereau, 75014 Paris, FRANCE</address>
                <email>vo.paris@obspm.fr</email>
                <telephone>+33140512089</telephone>
            </contact>
        </curation>
        <content>
            <subject>Registry</subject>
            <subject>Virtual Observatory</subject>
            <description>This registry provides access to resources
from VO-Paris Data Center and
                partners.</description>
            <referenceURL>http://vo.obspm.fr</referenceURL>
            <type>Registry</type>
            <contentLevel>General</contentLevel>
        </content>
        <rights>public</rights>
        <capability standardID="ivo://ivoa.net/std/Registry"
xsi:type="vg:Harvest">
            <interface role="std" version="1.0" xsi:type="vg:OAIHTTP">
                <accessURL>http://voparis-
registry.obspm.fr/oai/</accessURL>
            </interface>
            <maxRecords>0</maxRecords>
        </capability>
        <full>false</full>
        <managedAuthority>vopdc</managedAuthority>
        <managedAuthority>vopdc.obspm</managedAuthority>
        <managedAuthority>vopdc.iap</managedAuthority>
    </ri:Resource>
</ri:VOResources>
```

Appendix A.2 – Search XML response with identifiersOnly

```
http://example.com/registry/1/voresources/search?
keywords=vopdc&max=2&identifiersOnly
```

```
<ri:VOResources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
    xsi:schemaLocation="http://www.ivoa.net/xml/RegistryInterface/v1.0
http://www.ivoa.net/xml/RegistryInterface/v1.0"
    numberReturned="2" more="true" from="1">
    <ri:identifier>ivo://vopdc</ri:identifier>
    <ri:identifier>ivo://vopdc/registry</ri:identifier>
</ri:VOResources>
```