Virtual Observatory
Paris Data Centre

Full searchable registry REST interface

Jonathan Normand, Pierre Le Sidaner

Observatoire de Paris

Context

- **DAL Validator (Nara, Dec. 2010)**
- **IVOA Services Validator (Naples, May 2011)**
  - Goal: build a metric of IVOA services health
  - Steps:
    - Construct SOAP request
    - Send query to selected registry via WS
    - For each resource found examine services
    - For each service of xsi:type ParamHTTP perform validation
  - Issues:
    - Found duplicated resources managed by different registries
    - Found ghost resources (deletion issue)
    - Content of full searchable registries differs
  - ➡ Need a clean set of resources
- **Development of VOPDC full searchable registry (Heidelberg, May 2013)**

Overview

- **Database** (couchdb)
  - Get resources from publishing registries (OAI)
  - Keep only active and no duplicated resources
  - Convert subset of XML to JSON and ingest JSON document
- **Search engine** (elasticsearch)
  - Use Apache Lucene text-search engine
  - Only relevant fields are indexed
- **REST interface**
  - KISS principle
  - Easy to use
  - Lightweight and fast responses
  - No need to know SOAP and ADQL to query the registry
  - Allow simple or powerful queries (Lucene)
  - JSON representation

REST API Capabilities

- **Full API documentation and examples**

  `http://api.voparis-tmp.obspm.fr/registry/`

- **3 operations**

Search

Returns metadata of active resources matching a specific set of constraints

Get resource metadata

Returns metadata for a given resource

Get registry metadata

Returns metadata for this registry

Search operation

- **HTTP GET `/voresources/search`**
- **Set constraints on specific fields or on all fields**
- **Handle pagination mechanism**

- **Parameters**
  - `keywords` : set of constraints
  - `orValues` : combine multiple constraints with a logical OR
  - `from` : minimum position in the complete list of matching resources
  - `max` : maximum number of resources to return
  - `identifiersOnly` : get only resource identifiers.
- **Output**
  - JSON-formatted list of metadata for any resources

keywords syntax

- URL-encoded string
- Syntax follows the Apache Lucene query language
- Allow to set constraints on all fields of the JSON

ABNF notation

keywords = "keywords" "=" rules *(LWS rules)

rules = field-query | term-query

field-query = field-name ":" term-query

term-query = quoted-string

field-name = "identifier" | "description" | "title" | "type" | "subjects" | "publisher"
| "shortname" | "waveband" | "standardid"

keywords examples

- keywords=”infrared”
- keywords=standardid:"ivo://ivoa.net/std/SIA"
- keywords=standardid:"ivo://ivoa.net/std/ConeSearch" publisher:"cds"
- keywords=standardid:"ivo://ivoa.net/std/SSA" "galaxies"
- keywords=provenance:"ivo://archive.stsci.edu/nvoregistry"
- Another Lucene capablity:
  - keywords=updated:[2013 TO *]

Get resource operation

- **HTTP GET `/voresource` or `/voresource.xml`**

- **Parameters**
  - `identifier` : IVOA identifier

- **Outputs**
  - JSON-formatted list of metadata of the resource
  - XML VOResource.

- **Examples**
  - `http://voparis-registry.obspm.fr/vo/ivoa/1/voresources?identifier=ivo://vopdc`
  - `http://voparis-registry.obspm.fr/vo/ivoa/1/voresources.xml?identifier=ivo://vopdc`

Get registry operation

- **HTTP GET `/identity`**

- **No parameter**
  - Same as Get resource operation where IVOA identifier is the ID of this registry

- **Output**
  - JSON-formatted list of metadata of this registry resource.

- **Example**
  - `http://voparis-registry.obspm.fr/vo/ivoa/1/identity`

Current status and evolutions

- **Current status**
  - The REST API is fully functional
  - Need to schedule daily harvesting of publishing registries
- **Evolution of the API**
  - Easy to add new capabilities (add new endpoints)
  - Easy to extend metadata of resources (schema-less document)