International

Virtual

Observatory

Alliance

REST in the VO

Version 0.10

IVOA Working Draft 2008 May 05

This version:
http://www.ivoa.net/Documents/WD/GWS/VOREST-20080505.doc
Latest version:
http://www.ivoa.net/Documents/latest/VOREST.html
Previous version(s):
…
Author(s):
André Schaaff, Norman Gray
IVOA Grid and Web Services Working Group

Abstract

In the early years of the VO, the SOAP Web Service paradigm was an important element of the IVOA Architecture. Developments around these services are more and more complex with an increasing number of standards (WS-* …). REST [3] is not a standard but a formalization of the URL use and it is very easy to implement it. A service is RESTful if it follows a set of rules (which are not defined in a standard document). As there is no standard we think that it is necessary to define a minimal guideline about the “RESTfullness” in the VO context.

Status of This Document

This is an IVOA Working Draft. The first release of this document was 2008 May 05.

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

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

Acknowledgements

This document derives from discussions among the Grid and Web Services working group of the IVOA in Beijing (May 2007) and Cambridge (September 2007).

Contents

1 Introduction

REST (Representational State Transfer) [3] is not a standard but a formalization of the URL use. It was introduced in 2000 in Roy Fielding's Ph.D thesis [3]. It refers to a set of network architecture principles which describe how to define and address resources. There is no REST [3] standardization process and the term "REST" is often used when the service is not based on standards like SOAP.

A service is RESTful if it follows a small set of rules (which are not defined in a standard document). As there is no standard we think that it is necessary to define a minimal guideline about the "RESTfullness" in the VO context.

2 Resource Oriented Architecture [9]

A resource Oriented Architecture can be resumed to 4 concepts (resources, theirs names (URIs), their representation, the links between them) and 4 properties (addressability, statelessness, connectedness, uniform interface).

3 Representational State Transfer

3.1 Quick definition of REST and RESTful

In the REST approach, it may be sufficient to know the URI to access to a resource.

Examples:

1) http://www.example.com/sky/m31/pictures

2) http://www.example.com/sky/m31/picture/1

In these examples it is possible to access to the information through a simple URL without the use of a specific tool (for C#, Java, Perl …) on the client side, the client has just to read simply the URL.
In example 1) the URL returns the number of available pictures for m31 and in example 2) the URL returns the first picture.

Main feature of a REST service, from [9]:

Architectural style of the Web.
Resources are addressable (URIs).
Interact with representations of resources.
State is maintained within a resource representation.

Small set of methods that can be applied to any resource (HTTP methods).
Scaleable, low cost of coordination.

To be RESTful a service must be compliant with the following principles:

- Addressability

- Stateless

- Connectivity

- Uniform interface

3.2 Quick comparison with SOAP services

If we take again the two previous examples of REST [3] URLs,

1) http://www.example.com/sky/m31/pictures

2) http://www.example.com/sky/m31/picture/1

In SOAP we will have to define something like int getPictures (String object) for 1) and to use for example the SOAP with an attachment mechanism or to return the URL of the image.

SOAP Web Service engines are also evolving by implementing the REST [3] alternative. For example, in Axis 2 it is possible for a client to specify that we want to access the service following the REST [3] paradigm.

3.3 How to describe a service?

WSDL (Web Services Description Language) [7] has been created to describe SOAP services. These services are self-described by interrogation of the service endpoint URL with an extension like “?wsdl”. It is very difficult to use a SOAP Web Service without this description like when you try to the use a Java API without the corresponding Javadoc. For REST [3], there is no standard way like WSDL [7] but it is possible to use for example WADL (Web Application Description Language) [8]. RESTful services have simpler interfaces and the description is not as important as for SOAP WS. But in the case of automatic creation or use of the service by tools it is necessary to provide a description of the service. If a WADL [8] description is available it is then possible to generate for example the client Java code to query the service.

4 REST oriented frameworks and tools

4.1 Ruby on Rails [4]

Rails is a framework written in Ruby and dedicated to Web developments.
See [4] for more details.

4.2 Restlet [5]

Reslet is a framework for the Java platform providing native REST [3] support.
See [5] for more details.

4.3 Django [6]

Django is an open source framework for the Python expected to provide a native REST [3] support in the coming months.
See [6] for more details.

4.4 Comment

REST development or compliant frameworks are evolving quickly so we think that it we have not to recommend one or two of them.

5 Restfulness in the VO

5.1 Interoperability problems

As said in a previous part of this document, REST [3] is not a standard. The SOAP approach which was a key element in the IVOA architecture is a standard but is also crushed under a huge stack of related standards (WS-*).
As a minimum it would be very efficient to have a “standardized” description of the REST [3] services provided in the frame of the IVOA.
It could also be useful to have a tool to check if the service is really RESTful.

Use of an non-standard way (WADL for example)? Define our own standard?
Interaction with other WG (Registry…)?

Must be discussed in the frame of the GWS.

5.2 Recommendations

Add general recommendations ?
Must be discussed in the frame of the GWS.

5.3 Work to done

Continue this work as a profile for the next interop?
People are welcome from outside the VOTECH project to participate to this work.
Must be discussed in the frame of the GWS.

6 Conclusion

Compared to SOAP Web Services REST [3] is a lite way to provide a Web service following just a reduced set of rules. But it is perhaps necessary to define clearly the basis of what an interoperable RESTful VO service must be. REST [3] is more human oriented than SOAP but it defines no standard concerning the description of the service which is important in the case of dynamic use by other services.

Appendix A: “Appendix Title”

…

References

[1] R. Hanisch, *Resource Metadata for the Virtual Observatory* , http://www.ivoa.net/Documents/latest/RM.html
[2] R. Hanisch, M. Dolensky, M. Leoni, *Document Standards Management: Guidelines and Procedure* , http://www.ivoa.net/Documents/latest/DocStdProc.html
[3] R. Fielding, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
[4] Ruby on Rails, http://www.rubyonrails.org/
[5] Restlet, http://www.restlet.org/
[6] Django, http://www.djangoproject.com/
[7] WSDL, http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/
[8] WADL, https://wadl.dev.java.net/
[9] RESTful Web Services, L. Richardson and S. Ruby, O’Reilly