

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 6, 2011

S. Sheng
F. Arias
ICANN
June 4, 2011

A RESTful Web Service for Domain Name Registration Data (RWS-DNRD)
draft-sheng-weirds-icann-rws-dnrd-00

Abstract

This document describes a pilot RESTful Web Service for querying Domain Name Registration Data (aka WHOIS data).

The purpose of this document is to facilitate discussion and serve as input into a standards process in this area, currently being discussed on the WHOIS-based Extensible Internet Registration Data Service (WEIRDS) mailing list (<https://www.ietf.org/mailman/listinfo/weirds>).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 6, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Domain Name Registration Data	3
1.2. REST and RESTful Web Service	3
1.3. Why RESTful?	3
2. Terminology	4
3. Implementation Description	5
3.1. The Request	5
3.2. The Response	5
4. Error Codes	9
5. Formal XML Syntax	9
5.1. Contact XML Schema	11
5.2. Domain Name XML Schema	12
5.3. Host XML Schema	14
5.4. RWS XML Schema	15
6. Internationalization Considerations	16
7. IANA Considerations	16
8. Security Considerations	16
8.1. URIs and IRIs	16
9. Acknowledgments	16
10. References	16
10.1. Normative References	16
10.2. Informative References	17
Authors' Addresses	17

1. Introduction

This document describes a pilot implementation by ICANN staff for querying domain name registration data through a RESTful Web-based Interface. The service is implemented using the HTTP protocol [RFC2616] and conforms to the architectural constraints of REST [REST].

1.1. Domain Name Registration Data

Domain Name Registration Data are data that a registrant provides when s/he acquires or is assigned a domain name. By contract with the Internet Corporation for Assigned Names and Numbers (hereinafter ICANN), domain name system registries - domain registries and registrars as defined by [RFC3707] - are to provide public access to some of these data both via the WHOIS protocol ([RFC3912]) as well as via a web interface.

1.2. REST and RESTful Web Service

REST stands for Representational State Transfer. It is a set of architectural constraints that is developed as an abstract model of the Web architecture. These constraints include: client-server model, stateless, cacheable, layered system, code on demand (optional), and uniform interface. REST was used to guide the redesign of the Hypertext Transfer Protocol (HTTP) and Uniform Resource Identifiers. It is widely regarded as the architecture of the Web today. Principles of REST have been used to design other protocols such as the ATOM publishing protocol.

A RESTful web service is a web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects: 1) The "verbs" of the service are strictly those defined by the HTTP methods GET, PUT, POST, and DELETE, 2) The "verbs" are used to act upon resources, and 3) resources are addressable using URLs.

1.3. Why RESTful?

Compared to the WHOIS protocol as defined in RFC 3912, we felt the RESTful approach offers the following advantages:

Standardized output and error format: The base response output format is XML, which when paired with a well-defined schema would allow for automated processing.

Support for internationalisation: RWS has complete support for internationalised registration data, as well as IDNs with

U-labels, by using the XML data format, which uses Unicode.

Authentication and access control: HTTP, the transport for RWS, already supports authentication, and by means of using these capabilities, RWS makes technically possible to implement granular permissions over registration data if required.

Addressable Whois Service: RWS requires the use of a URI/URL standard structure for each object/resource. This has the additional benefit of providing a widely recognized manner to refer unambiguously to objects in Whois.

Increased Usability: Some of the inherent capabilities of the HTTP protocol (such as redirects) can be used to provide additional functionality such as automatic referrals to more specific WHOIS data sources without requiring specialized parsing by the client.

Authenticity of Origin: RWS provided over HTTPS offers confidence in the origin of the information.

Leverage existing infrastructure and expertise: RWS is HTTP-based and can be supported using popular web server infrastructures. Web administration is a skill-set and resource likely already commonplace inside registries and registrars. Similarly, RWS can benefit from existing technology to implement load-balance servers, cache answers to minimize network traffic, etc.

2. Terminology

For convenience, this implementation can be referred to as the "RESTful Web Service for Domain Name Registration Data" or "RWS - DNRD". The following terminology is used by this specification:

Domain Name Registration Data (aka WHOIS data) - refers to the data that registrants provide when registering a domain name. Part of these data are sometimes contractually required to be made publicly available, these includes the name of the registrant, name servers, the expiration date of the registration, etc. (see section 3.3 section of Registrar Accreditation Agreement).

URI - A Uniform Resource Identifier as defined in [RFC3986].

IRI - An internationalized Resource Identifier as defined in [RFC3987].

Resource - A network-accessible data object or service identified by an URI, as defined in [RFC2616]. In this context, resources

refers to the registration data objects.

Representation - An entity included with a request or response as defined in [RFC2616].

3. Implementation Description

ICANN's RWS-DNRD specifies the URL structure, the methods to be used, the responses and its format, and the result codes.

As its name implies RWS-DNRD is Web-based, i.e., uses HTTP [RFC2616] as its transport. Given its RESTful nature it only uses the standard HTTP methods. And given it is read-only, it only uses the GET and HEAD methods.

3.1. The Request

The server accepts standard HTTP "GET" requests for the resources it serves. The client sends its request with the following URI structure.

3.1.1. URL structure

Meta Data - URL: / Display information about resources, URI scheme and allowed operations

Domain Name Request - /domain/<name>/ example:
http://whois.test/domain/example.test/

Contact Request - /contact/<id>/ example:
http://whois.test/contact/CID-4005/

Host Request - /host/<name>/ example:
http://whois.test/host/ns1.example.test/

3.2. The Response

The RWS-DNRD server provides responses in XML format as specified below and can offer responses in other formats; currently only HTML and plain text. The client signals the preferred format using the standard HTTP "Accept:" header. The client can also signal the preferred format by adding a DOS-file-style extension to the resource. For example, "/xreg/rsrcl.xml". If the client specifies no preferred format, the server responds in XML. If the client signals one format with the HTTP "Accept:" header and another with the extension style, the server ignores the preference signaled with the former.

The pilot implementation currently support the following values for the Accept header: application/xml for XML, text/html for HTML, and text/plain for plain text.

The root element for a RWS-DNRD response is <rws>. This element contains one or more <result> elements that are explained in the following section.

Example of root element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rws:rws xmlns:rws="urn:ietf:params:xml:ns:rws-1.0"
  ...
  <rws:result>
  ...
  </rws:result>

  <rws:result>
  ...
  </rws:result>
  ...
</rws:rws>
```

3.2.1. Response for Domain Names

Example Query: <http://whois.test/domain/example.test/>

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rws xmlns="urn:ietf:params:xml:ns:rws-1.0"
  xmlns:rws="urn:ietf:params:xml:ns:rws-1.0">
  <result>
    <rws:domain xmlns="urn:ietf:params:xml:ns:rwsDomain-1.0"
      xmlns:rwsDomain="urn:ietf:params:xml:ns:rwsDomain-1.0">
      <name>example.test</name>
      <roid>9690-TEST</roid>
      <status s="clientHold"/>
      <status s="clientRenewProhibited"/>
      <status s="clientUpdateProhibited"/>

      <registrant href="/contact/4447">4447</registrant>
      <contact type="admin" href="/contact/4447">4447</contact>
      <contact type="tech" href="/contact/4447">4447</contact>
      <ns>
        <hostObj href="/host/ns1.example.test">
          ns1.example.test
        </hostObj>
        <hostObj href="/host/ns2.example.test">
          ns2.example.test
        </hostObj>
        <hostObj href="/host/ns3.example.test">
          ns3.example.test
        </hostObj>
      </ns>
      <clID>793</clID>
      <clName>XYZ Corporation</clName>
      <crID>1289</crID>
      <crDate>1992-07-26T09:10:56Z</crDate>

      <exDate>2019-01-21T10:11:18Z</exDate>
    </rws:domain>
  </result>
</rws>
```

3.2.2. Response for Contacts

Example Query: <http://whois.test/conact/4447/>

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rws xmlns="urn:ietf:params:xml:ns:rws-1.0"
  xmlns:rws="urn:ietf:params:xml:ns:rws-1.0">
  <result>
    <rws:contact xmlns="urn:ietf:params:xml:ns:rwsContact-1.0"
      xmlns:rwsContact="urn:ietf:params:xml:ns:rwsContact-1.0">
      <id>4447</id>
      <roid>4447-TEST</roid>
      <status s="clientDeleteProhibited"/>
      <status s="clientTransferProhibited"/>
      <status s="ok"/>

      <rwsContact:postalInfo
        xmlns="urn:ietf:params:xml:ns:contact-1.0" type="int">
        <name>Nova D Janes</name>
        <addr>
          <street>1755 XYZ Avenue</street>
          <city>ABC</city>
          <sp>PA</sp>
          <pc>15206</pc>
          <cc>US</cc>
        </addr>
        </rwsContact:postalInfo>
        <voice>+1.1234567890</voice>
        <fax>+1.1234567890</fax>
        <email>Nova.D.Janes@xyz.com</email>
        <clID>1289</clID>
        <crID>1289</crID>
        <crDate>1995-11-29T03:17:09Z</crDate>
        <trDate>1997-02-10T21:56:43Z</trDate>
      </rws:contact>
    </result>
  </rws>
```

3.2.2.1. Response for Host Names

Example Query: <http://whois.test/host/ns1.example.test/>

Response:


```
<?xml version="1.0" encoding="UTF-8"?>
<rws xmlns="urn:ietf:params:xml:ns:rws-1.0"
  xmlns:rws="urn:ietf:params:xml:ns:rws-1.0">
  <result>
    <rws:host xmlns="urn:ietf:params:xml:ns:rwsHost-1.0">
      <name>ns1.example.test</name>
      <roid>8998-TEST</roid>
      <status s="ok"/>
      <status s="clientDeleteProhibited"/>
      <status s="serverDeleteProhibited"/>

      <addr ip="v4">188.17.219.3</addr>
      <clID>1289</clID>
      <crID>1289</crID>
      <crDate>1995-01-12T01:26:27Z</crDate>
      <upID>1289</upID>
      <upDate>2005-03-01T08:43:11Z</upDate>

    </rws:host>
  </result>
</rws>
```

4. Error Codes

In compliance with the REST paradigm any error information is returned in the form of a standard HTTP response with an HTTP status code describing the error and a text/plain body message describing the exception causing the error response. In this version we are using only standard HTTP codes (<http://www.iana.org/assignments/http-status-codes>).

We plan to define specialized error codes in the future.

5. Formal XML Syntax

The formal syntax presented here is a complete schema representation suitable for automated validation of an XML instance. We base our implementation on the Extension Provisioning Protocol (EPP). We reference and include the following EPP schemas:

[RFC5730] - Extensible Provisioning Protocol (EPP)

[RFC5731] - Extensible Provisioning Protocol (EPP) Domain Name Mapping

[RFC5732] - Extensible Provisioning Protocol (EPP) Host Mapping

[RFC5733] - Extensible Provisioning Protocol (EPP) Contact Mapping

5.1. Contact XML Schema

```
<schema targetNamespace="urn:ietf:params:xml:ns:rwsContact-1.0"
  elementFormDefault="qualified">

  <!--
  Import common element types.
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:rws-1.0" />
  <import namespace="urn:ietf:params:xml:ns:contact-1.0" />

  <annotation>
    <documentation>
      RESTful Whois schema for contact.
    </documentation>
  </annotation>

  <!--
  Child elements of Contact object
  -->
  <complexType name="rwsContactType">
    <sequence>
      <element name="id" type="eppcom:clIDType"/>
      <element name="roid" type="eppcom:roidType"/>
      <element name="status" type="contact:statusType" maxOccurs="7"/>
      <element name="postalInfo" type="contact:postalInfoType"
        maxOccurs="2"/>
      <element name="voice" type="contact:e164Type" minOccurs="0"/>
      <element name="fax" type="contact:e164Type" minOccurs="0"/>
      <element name="email" type="eppcom:minTokenType"/>
      <element name="clID" type="eppcom:clIDType"/>
      <element name="crID" type="eppcom:clIDType"/>
      <element name="crDate" type="dateTime"/>
      <element name="upID" type="eppcom:clIDType" minOccurs="0"/>
      <element name="upDate" type="dateTime" minOccurs="0"/>
      <element name="trDate" type="dateTime" minOccurs="0"/>
      <element name="extension" type="rws:extAnyType" minOccurs="0"/>
    </sequence>
  </complexType>

  <!--
  End of schema.
  -->
</schema>
```

5.2. Domain Name XML Schema

```
<schema targetNamespace="urn:ietf:params:xml:ns:rwsDomain-1.0"
  elementFormDefault="qualified">

  <!--
  Import common element types
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0" />
  <import namespace="urn:ietf:params:xml:ns:domain-1.0" />
  <import namespace="urn:ietf:params:xml:ns:contact-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rws-1.0" />

  <annotation>
    <documentation>
      RESTful Whois schema for domains
    </documentation>
  </annotation>

  <!--
  Child elements of a Domain object
  -->
  <complexType name="rwsDomainType">
    <sequence>
      <element name="name" type="eppcom:labelType" />
      <element name="uName" type="eppcom:labelType" minOccurs="0" />
      <element name="roid" type="eppcom:roidType" />
      <element name="status" type="domain:statusType"
        maxOccurs="unbounded" />
      <element name="registrant" type="rwsDomain:registrantType"
        minOccurs="0" />
      <element name="contact" type="rwsDomain:contactType"
        minOccurs="0" maxOccurs="unbounded" />
      <element name="ns" type="rwsDomain:nsType" minOccurs="0" />
      <element name="host" type="rwsDomain:hostObjType"
        minOccurs="0" maxOccurs="unbounded" />
      <element name="clID" type="eppcom:clIDType" />
      <element name="clName" type="contact:postalLineType" />
      <element name="crID" type="eppcom:clIDType" minOccurs="0" />
      <element name="crDate" type="dateTime" minOccurs="0" />
      <element name="upID" type="eppcom:clIDType" minOccurs="0" />
      <element name="upDate" type="dateTime" minOccurs="0" />
      <element name="exDate" type="dateTime" minOccurs="0" />
      <element name="trDate" type="dateTime" minOccurs="0" />
      <element name="extension" type="rws:extAnyType"
        minOccurs="0" />
    </sequence>
  </complexType>

```

```
    </sequence>
  </complexType>

  <complexType name="registrantType">
    <simpleContent>
      <extension base="eppcom:clIDType">
        <attribute name="href" type="anyURI"/>
      </extension>
    </simpleContent>
  </complexType>

  <complexType name="contactType">
    <simpleContent>
      <extension base="eppcom:clIDType">
        <attribute name="type" type="domain:contactAttrType"/>
        <attribute name="href" type="anyURI"/>
      </extension>
    </simpleContent>
  </complexType>

  <complexType name="nsType">
    <choice>
      <element name="hostObj" type="rwsDomain:hostObjType"
        maxOccurs="unbounded"/>
      <element name="hostAttr" type="domain:hostAttrType"
        maxOccurs="unbounded"/>
    </choice>
  </complexType>

  <complexType name="hostObjType">
    <simpleContent>
      <extension base="eppcom:labelType">
        <attribute name="href" type="anyURI"/>
      </extension>
    </simpleContent>
  </complexType>

  <!--
    End of schema.
  -->
</schema>
```

5.3. Host XML Schema

```
<schema targetNamespace="urn:ietf:params:xml:ns:rwsHost-1.0"
  elementFormDefault="qualified">

  <!--
  Import common element types
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rws-1.0" />
  <import namespace="urn:ietf:params:xml:ns:host-1.0" />

  <annotation>
    <documentation>
      RESTful Whois schema for hosts
    </documentation>
  </annotation>

  <!--
  Child elements of Host object
  -->
  <complexType name="rwsHostType">
    <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="roid" type="eppcom:roidType"/>
      <element name="status" type="host:statusType" maxOccurs="7"/>
      <element name="addr" type="host:addrType" minOccurs="0"
        maxOccurs="unbounded"/>
      <element name="clID" type="eppcom:clIDType"/>
      <element name="crID" type="eppcom:clIDType"/>
      <element name="crDate" type="dateTime"/>
      <element name="upID" type="eppcom:clIDType" minOccurs="0"/>
      <element name="upDate" type="dateTime" minOccurs="0"/>
      <element name="trDate" type="dateTime" minOccurs="0"/>
      <element name="extension" type="rws:extAnyType" minOccurs="0"/>
    </sequence>
  </complexType>

  <!--
  End of schema.
  -->
</schema>
```

5.4. RWS XML Schema

```
<schema targetNamespace="urn:ietf:params:xml:ns:rws-1.0"
  elementFormDefault="qualified">

  <!--
    Import common element types
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rwsContact-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rwsHost-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rwsDomain-1.0" />

  <annotation>
    <documentation>
      RESTful Whois schema
    </documentation>
  </annotation>

  <!--
    Root element
  -->
  <element name="rws" type="rws:rwsType"/>
  <attribute name="uri" type="anyURI"/>

  <!--
    RWS types
  -->
  <complexType name="rwsType">
    <sequence>
      <element name="result" type="rws:resultType" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="resultType">
    <choice>
      <element name="domain" type="rwsDomain:rwsDomainType"/>
      <element name="contact" type="rwsContact:rwsContactType"/>
      <element name="host" type="rwsHost:rwsHostType"/>
      <element name="extension" type="rws:extAnyType"/>
    </choice>
  </complexType>

  <!--
    Extension framework type
  -->
  <complexType name="extAnyType">
    <sequence>
      <any namespace="##other" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</schema>
```

```
    </sequence>
  </complexType>

  <!--
    End of schema.
  -->
</schema>
```

6. Internationalization Considerations

Information published in RWS is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an `<?xml?>` declaration, use of UTF-8 is preferred.

7. IANA Considerations

TBD

8. Security Considerations

8.1. URIs and IRIs

RWS-DNRD implementations use URIs and IRIs. See Section 7 of [RFC3986] and Section 8 of [RFC3987] for security considerations related to their handling and use.

9. Acknowledgments

The authors would like to acknowledge the following individuals for their input: Andrew Sullivan, and Dave Piscitello.

10. References

10.1. Normative References

[REST] Fielding, R. and R. Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology Vol. 2, No. 2, May 2002.

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.
- [RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, August 2009.
- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, August 2009.

10.2. Informative References

- [RFC3707] Newton, A., "Cross Registry Internet Service Protocol (CRISP) Requirements", RFC 3707, February 2004.
- [RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, September 2004.

Authors' Addresses

Steve Sheng
Internet Corporation for Assigned Names and Numbers
4676 Admiralty Way, Suite 330
Marina del Rey 90292
United States of America

Phone: +1.310.823.9358
Email: steve.sheng@icann.org

Francisco Arias
Internet Corporation for Assigned Names and Numbers
4676 Admiralty Way, Suite 330
Marina del Rey 90292
United States of America

Phone: +1.310.823.9358
Email: francisco.arias@icann.org

