Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: November 8, 2012

S. Hollenbeck
Verisign Labs
S. Sheng
F. Arias
ICANN

May 7, 2012

Domain Name Registration Data Access Protocol Query Format
draft-hollenbeck-dnrd-ap-query-01

Abstract

This document describes a RESTful query format proposal for the Domain Name Registration Data Access Protocol.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions Used in This Document 3
  2.1. Acronyms and Abbreviations 3
3. Design Considerations 4
  3.1. Why RESTful? 4
4. Protocol Specification 5
  4.1. Base URL Specification 6
  4.2. Domain Path Segment Specification 6
  4.3. Name Server Path Segment Specification 7
  4.4. Contact Path Segment Specification 7
  4.5. Registrar Name Path Segment Specification 8
  4.6. Response Preference Specification 8
5. Query Parameters 9
6. Client Identification 10
7. Internationalization Considerations 10
  7.1. Label Considerations 10
  7.2. Label Encoding 10
8. IANA Considerations 11
9. Security Considerations 11
10. Acknowledgements 11
11. References 11
  11.1. Normative References 11
  11.2. Informative References 12
Authors' Addresses 13

1. Introduction

This document describes a specification for querying domain name registration data using a RESTful web service and uniform query patterns. The service is implemented using the Hypertext Transfer Protocol (HTTP) [RFC2616] and conforms to the architectural constraints of Representational State Transfer (REST) [REST].

The protocol described in this specification is intended to address deficiencies with the WHOIS protocol [RFC3912] that have been identified over time, including:

Lack of standardized command structures,

lack of standardized output and error structures,

lack of support for internationalization and localization, and

lack of support for user identification, authentication, and access control.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The terms "registry", "registrar", and "registrant" are to be interpreted as described in RFC 3707 [RFC3707].

2.1. Acronyms and Abbreviations

DNRD: Domain Name Registration Data

HTTP: Hypertext Transfer Protocol, specified in RFC 2616 [RFC2616]

HTTP/TLS: HTTP over TLS, specified in RFC 2818 [RFC2818]

IDN: Internationalized Domain Name, specified in RFC 5890 [RFC5890]

JSON: JavaScript Object Notation, based on a subset of the JavaScript Programming Language standard [ECMA]

REST: Representational State Transfer [REST]

RWS: RESTful Web Service

TLS: Transport Layer Security, specified in RFC 5246 [RFC5246]

URI: Uniform Resource Identifier, specified in RFC 3986 [RFC3986]

URL: Uniform Resource Locator, specified in RFC 3986 [RFC3986]

XML: Extensible Markup Language, specified in W3C Recommendation REC-xml-20081126 [W3C.REC-xml-20081126]

3. Design Considerations

Representational State Transfer (REST) is a style of software architecture for distributed systems. The style describes six constraints: client-server, stateless, cacheable, layered system, code on demand (optional), and uniform interface. Systems that comply with these constraints are designed to have the properties of performance, scalability, simplicity, modifiability, visibility, portability, and reliability. The principles of REST have been used to design other protocols such as the ATOM publishing protocol [RFC5023].

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:

o The "verbs" of the service are those strictly defined by the HTTP methods GET, PUT, POST, and DELETE,

o the "verbs" are used to act upon resources, and

o resources are addressable using URLs.

3.1. Why RESTful?

A RESTful approach to querying domain registration data offers several advantages when compared to the WHOIS protocol, including:

Standardized output and error structures: outputs can be structured using encoding technologies like JSON and XML, which when paired with a well-defined specification will allow for automated processing.

Support for internationalization: RWS structured data formats include complete support for both internationalized registration data and Internationalized Domain Names (IDNs) with U-labels.

Authentication and access control: HTTP, the transport for RWS, supports multiple native user identification and authentication schemes, and by using these capabilities RWS makes it possible to implement registration data access control mechanisms.

Addressable service: RWS requires the use of a URI/URL standard structure for each object/resource. This provides a way to unambiguously refer to objects.

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

Authenticity of origin: RWS provided over HTTP/TLS provides confidence in the origin of the information.

Leverage existing infrastructure and expertise: RWS is HTTP-based and is supported using popular, commonly deployed web server infrastructures.

4. Protocol Specification

This section describes the DNRD-AP URL structure and methods used to create the uniform patterns needed to submit queries over HTTP. Each query is sent to the server in the form of an HTTP "GET" or HTTP "HEAD" request. A "GET" request will return both response headers and a response body. A "HEAD" request will return only response headers. A "HEAD" request can be used to verify URL syntax or resource availability without actually retrieving the requested resource.

General specifications for using HTTP in a system to provide a RESTful DNRD query service are described in X (the design team HTTP draft).

Client-supplied parameters MUST be interpreted in a case-insensitive, exact match manner in order to produce no more than one matching result. Client parameters that can be used to match multiple registry or registrar data elements are described in (the domain search draft).

4.1. Base URL Specification

The uniform patterns start with a base URL [RFC3986] specified by the service provider offering this service. Resource-type specific path segments are then appended to the end of the base URL. The base URL may contain its own path segments (e.g. http://example.com/... or http://example.com/dnrd-ap/...).

The resource types that can be used to construct path segments include:

'domain': Used to identify a domain name query.

'nameserver': Used to identify a name server query.

'contact': Used to identify a contact query.

'registrar': Used to identify a registrar name query.

The resource type MAY be omitted from the path segment. If the resource type is omitted, the path segment MUST be interpreted as a domain path segment.

4.2. Domain Path Segment Specification

Syntax: domain/<domain name> or <domain name>

The <domain name> parameter represents a domain name as specified in RFC 4343 [RFC4343]. Internationalized domain names represented in both A-label and U-label formats [RFC5890] are also valid domain names. The following URLs are example queries for domain name registration information:

```
http://example.com/dnrd-ap/domain/example.com/
http://example.com/dnrd-ap/example.com/
```

HTTP GET Request Format:

```
GET /dnrd-ap/domain/example.com HTTP/1.1
Host: example.com
```

HTTP HEAD Request Format:

```
HEAD /dnrd-ap/domain/example.com HTTP/1.1
```

```
      Host: example.com
```

4.3. Name Server Path Segment Specification

Syntax: nameserver/<name server name>

The <name server name> parameter represents a host name as specified in RFC 952 [RFC0952] and RFC 1123 [RFC1123]. Internationalized host names represented in A-label format [RFC5890] are also valid host names. The following URLs are example queries for name server registration information:

```
      http://example.com/dnrd-ap/nameserver/ns1.example.com/
```

HTTP GET Request Format:

```
      GET /dnrd-ap/nameserver/ns1.example.com/ HTTP/1.1
      Host: example.com
```

HTTP HEAD Request Format:

```
      HEAD /dnrd-ap/nameserver/ns1.example.com/ HTTP/1.1
      Host: example.com
```

4.4. Contact Path Segment Specification

Syntax: contact/<contact id>

The <contact id> parameter represents a contact identifier as specified in RFC 5730 [RFC5730] and RFC 5733 [RFC5733]. The following URL is an example query for contact registration information:

```
   http://example.com/dnrd-ap/contact/CID-4005/
```

HTTP GET Request Format:

```
      GET /dnrd-ap/contact/CID-4005/ HTTP/1.1
      Host: example.com
```

HTTP HEAD Request Format:

```
      HEAD /dnrd-ap/contact/CID-4005/ HTTP/1.1
      Host: example.com
```

4.5. Registrar Name Path Segment Specification

Syntax: registrar/<registrar name>

The <registrar name> parameter represents a Unicode text string as specified in RFC 5198 [RFC5198]. The following URL is an example query for registrar information:

```
http://example.com/dnrd-ap/registrar/Example Registrar, Inc./
```

HTTP GET Request Format:

```
GET /dnrd-ap/registrar/Example Registrar, Inc./ HTTP/1.1
Host: example.com
```

HTTP HEAD Request Format:

```
HEAD /dnrd-ap/registrar/Example Registrar, Inc./ HTTP/1.1
Host: example.com
```

4.6. Response Preference Specification

DNRD-AP servers return responses encoded using one of multiple algorithms. The client MAY signal the preferred format using an HTTP "Accept:" header. The client can also signal the preferred format by adding a DOS-file-style extension to the resource. For example, "/domain/example.com.xml/". If the client specifies no preferred format the server MUST encode the response using a default format. If the client signals multiple formats with the HTTP "Accept:" header, or one format with the HTTP "Accept:" header and another with the extension style, the response will be encoded as described in Section X of (the draft DNRD-AP response document).

The following media type values can be specified with the "Accept:" header:

application/xml (for an XML-encoded response)

application/json (for a JSON-encoded response)

text/html (for an HTML-encoded response)

text/plain (for a plain text response)

HTTP GET Request Format for an XML-encoded Response:

```
GET /dnrd-ap/domain/example.com HTTP/1.1
Host: example.com
Accept: application/xml
```

HTTP HEAD Request Format for an XML-encoded Response:

```
HEAD /dnrd-ap/domain/example.com HTTP/1.1
Host: example.com
Accept: application/xml
```

Alternate HTTP GET Request Format for an XML-encoded Response:

```
GET /dnrd-ap/domain/example.com.xml HTTP/1.1
Host: example.com
```

Alternate HTTP HEAD Request Format for an XML-encoded Response:

```
HEAD /dnrd-ap/domain/example.com.xml HTTP/1.1
Host: example.com
```

HTTP GET Request Format for an XML- or JSON-encoded Response:

```
GET /dnrd-ap/domain/example.com HTTP/1.1
Host: example.com
Accept: application/xml,application/json
```

HTTP HEAD Request Format for an XML- or JSON-encoded Response:

```
HEAD /dnrd-ap/domain/example.com HTTP/1.1
Host: example.com
Accept: application/xml,application/json
```

5. Query Parameters

To overcome issues with misbehaving HTTP cache infrastructure, clients may use the '__dnrd__cachebust' query parameter with a random value of their choosing. Servers MUST ignore this query parameter.

The following is an example use of this parameter to retrieve the domain registration data for the example.com domain:

```
http://example.com/dnrd-ap/domain/example.com?__dnrd_cachebust=xyz123
```

Clients SHOULD NOT send any other query parameters.

6. Client Identification

Access to resources can be restricted to clients that possess identification credentials negotiated using an out-of-band mechanism. For example, a service provider can provide clients with user names and passwords as part of a service agreement to gain access to restricted resources. If available, clients MAY provide user name and password identification information to a server using the HTTP "basic" authentication scheme described in RFC 2617 [RFC2617]. Considerations for making authorization and access control decisions based on client-provided identification information are described in Section X of (the draft DNRD-AP response document).

Client user names and passwords MUST be protected using a facility that provides privacy and integrity services to protect against unintended disclosure and modification while in transit. At a minimum, support for HTTP/TLS as described in RFC 2818 [RFC2818] MUST be provided. Service providers can optionally specify and deploy additional security services.

7. Internationalization Considerations

7.1. Label Considerations

There is value in supporting the ability to submit either a U-label (Unicode form of an IDN label) or an A-label (ASCII form of an IDN label) as a query argument to a DNRD service. Users may most often prefer a U-label since this is more visually recognizable and familiar than A-label strings, but users of programmatic interfaces may wish to submit and display A-labels or may not be able to input U-labels with their keyboard configuration.

Internationalized domain and host names can contain character variants and variant labels as described in RFC 4290 [RFC4290]. Clients that support queries for internationalized domain and host names MUST accept service provider responses that describe variants as specified in (the draft DNRD-AP response document).

7.2. Label Encoding

Internationalized labels can be encoded in any of three different ways:

U-label only: A U-label is entered as part of a path segment. For example, /domain/"U+82F1""U+96C4".example.

A-label only: A U-label is first converted to its corresponding A-label before being submitted to the server. In the example above, the U-label would be converted to "xn--dj1az91b", and the path segment would be /domain/xn--dj1az91b.example.

IRI -> URI conversion: An IRI (which contains the U-label) is converted to a URI using the algorithm described in RFC 3987 [RFC3987] before being submitted to the server. In the example above, the label would be converted to "%E8%8B%B1%E9%9B%84" and the path segment becomes /domain/%E8%8B%B1%E9%9B%84.example.

8. IANA Considerations

This document does not specify any IANA actions.

9. Security Considerations

All of the security considerations described for HTTP in RFC 2616 [RFC2616], HTTP Basic Authentication in RFC 2617 [RFC2617], HTTP Over TLS in RFC 2818 [RFC2818], and their successors are applicable. There are no additional considerations introduced by this specification.

10. Acknowledgements

The authors would like to acknowledge the following individuals for their contributions to this document: Andrew Newton.

11. References

11.1. 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.

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3707] Newton, A., "Cross Registry Internet Service Protocol (CRISP) Requirements", RFC 3707, February 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4290] Klensin, J., "Suggested Practices for Registration of Internationalized Domain Names (IDN)", RFC 4290, December 2005.

[RFC4343] Eastlake, D., "Domain Name System (DNS) Case Insensitivity Clarification", RFC 4343, January 2006.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, August 2009.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

11.2. Informative References

[ECMA] European Computer Manufacturers Association, "ECMAScript Language Specification 3rd Edition", December 1999, <http:

//www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf>.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, September 2004.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

[RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.

[W3C.REC-xml-20081126] Sperberg-McQueen, C., Yergeau, F., Bray, T., Maler, E., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

Authors' Addresses

Scott Hollenbeck
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
US

Email: shollenbeck@verisign.com
URI: http://www.verisignlabs.com/

Steve Sheng
Internet Corporation for Assigned Names and Numbers
4676 Admiralty Way, Suite 330
Marina del Rey, CA 90292
US

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, CA 90292
US

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