**RESTful WHOIS**

**Reference Manual**

Table of Contents

[1. Introduction 3](#_Toc414818806)

[1.1. RDAP Modules 3](#_Toc414818807)

[2. Install 4](#_Toc414818808)

[2.1. Supported Operating Systems 4](#_Toc414818809)

[2.2. rdap-service 4](#_Toc414818810)

[2.3. rdap-proxy43 5](#_Toc414818811)

[3. Configuration 7](#_Toc414818812)

[3.1. Database configuration 7](#_Toc414818814)

[3.2. Common configuration 7](#_Toc414818815)

[3.3. Bootstrap configuration 8](#_Toc414818816)

[4. Query API 9](#_Toc414818817)

[4.1. Introduction 9](#_Toc414818819)

[4.2. Response Code 10](#_Toc414818820)

[4.3. Api 10](#_Toc414818821)

[5. Update API 12](#_Toc414818822)

[5.1. Introduction 12](#_Toc414818824)

[5.2. Common request format 13](#_Toc414818825)

[create 13](#_Toc414818826)

[update 13](#_Toc414818827)

[delete 13](#_Toc414818828)

[5.3. Common response format 13](#_Toc414818829)

[5.4. Response code 14](#_Toc414818830)

[5.5. Request body parameter 15](#_Toc414818831)

[Common parameter 15](#_Toc414818832)

[Domain 17](#_Toc414818833)

[Nameserver 20](#_Toc414818834)

[Entity 21](#_Toc414818835)

[Network 23](#_Toc414818836)

[As number 23](#_Toc414818837)

[6. Proxy43 24](#_Toc414818838)

[6.1. Introduction 24](#_Toc414818840)

[6.2. Api 24](#_Toc414818841)

[7. Customize and Develop 27](#_Toc414818842)

[7.1. Use Registry’s database 27](#_Toc414818844)

[7.2. Function pick 27](#_Toc414818845)

[7.3. Customize Validator 28](#_Toc414818846)

[7.4. Enable/disable access control 28](#_Toc414818847)

[7.5. Enable/disable redirect 28](#_Toc414818848)

[7.6. Add custom features 28](#_Toc414818849)

[7.7. VCARD extension 28](#_Toc414818850)

[8. Other 29](#_Toc414818851)

# Introduction

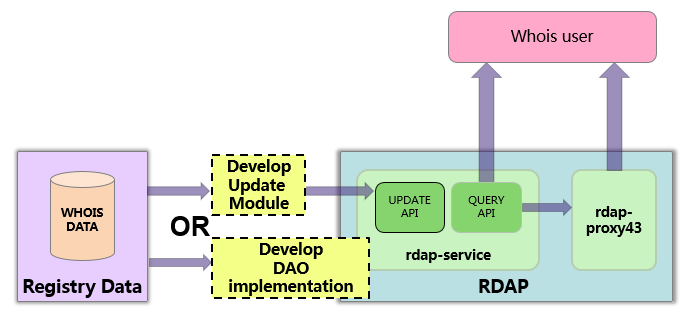
Restful WHOIS is a RESTful-based Registration Data Access Protocol ([RDAP protocol](http://datatracker.ietf.org/wg/weirds/)) implementation for Domain Registries and Regional Internet Registries.

This project is a starting point for registries to build restful WHOIS service so that they need not to start from scratch.

The term ‘RDAP’ will be used for this project name in following sections.

## RDAP Modules

The project is written in JAVA. Modules:



1. rdap-service

rdap-service Is a HTTP service, can be deployed in SERVLET container, such as Tomcat or Jetty. It loads data from MYSQL database by default

It has two kinds of API:

1. query API

Registrar user can query WHOIS data from this API.

1. update API

Registry can update WHOIS data from this API.

There are two approaches for registries to implement:

1. Update Restful Whois data via data update API.
2. Develop DAO module of rdap-service to access the DB of the registry.
3. rdap-proxy43

TCP 43 port service, a standalone server, the same function with old WHOISD.

# Install

## Supported Operating Systems

Tested operating environment:

1. Red Hat Enterprise Linux Server release 5.3
2. CentOS release 5.7
3. Win7
4. Win8
5. OS X 10.8.4.

## rdap-service

1. Install [JDK7](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html), or higher version. (Skip this step if already installed)
2. Install [Mysql5](http://dev.mysql.com/downloads/mysql), or higher version. (Skip this step if already installed)

Create user 'rdap' and grant privilege.

($RDAP\_SERVER\_IP must change to rdap server IP, $MYSQL\_PASSWORD change to 'rdap' user's password)

GRANT ALL PRIVILEGES ON \*.\* TO 'rdap'@'$RDAP\_SERVER\_IP' IDENTIFIED BY '$MYSQL\_PASSWORD';

FLUSH PRIVILEGES;

More details please ref [here](https://github.com/cnnic/rdap/wiki/%5Binstall%5D-Mysql-privilege).

1. Install Tomcat7, or higher version. (Skip this step if already installed)

[Download](http://tomcat.apache.org/download-70.cgi) and [Install Tomcat7](http://tomcat.apache.org/tomcat-7.0-doc/setup.html) or higher version, and HTTP port use default port 8080 (see [here](http://tomcat.apache.org/tomcat-7.0-doc/RUNNING.txt) if use other port).  
Installed Tomcat root folder called '$TOMCAT\_HOME', which contains folders:bin,conf,lib,webapps, etc.

1. Get RDAP war file.  
   There are two methods to get RDAP war file:
   * You can get [war file](https://github.com/cnnic/rdap/raw/dev/rdap-service/build/rdap-service-1.0.war) building by JDK7.
   * Or [Build war file from source](https://github.com/cnnic/rdap/wiki/%5Binstall%5DBuild-war-file-from-source)
2. Deploy RDAP war to tomcat.
   * Create folder 'rdap' in dir $TOMCAT\_HOME/webapps/
   * Unzip RDAP war file to $TOMCAT\_HOME/webapps/rdap/
   * Edit database configuration file [jdbc.properties](https://github.com/cnnic/rdap/wiki/jdbc.properties)
   * Edit global configuration file [rdap.properties](https://github.com/cnnic/rdap/wiki/rdap.properties)
3. Init database.  
   This step will create database named 'rdap', and you can insert test data into it.  
   This step will use database info in jdbc.properties you have configured before.  
   WARN: this step will DROP database of 'jdbc.url.dbName' if it is existing, and then recreate it.

cd $TOMCAT\_HOME/webapps/rdap/WEB-INF/classes

CLASSPATH=.:$CLASSPATH #in windows this command can be ignored

java -Djava.ext.dirs=../lib org.restfulwhois.rdap.init.Init initschema #DROP database 'jdbc.url.dbName', and recreate it, and create table.

1. Start up tomcat
   * Start up tomcat

[in Linux/OS X, open a shell and execute command:]

cd $TOMCAT\_HOME #$TOMCAT\_HOME must be replaced by real dir

bin/startup.sh

[in Windows, open command prompt window and execute command:]

cd $TOMCAT\_HOME/bin #$TOMCAT\_HOME must be replaced by real dir

startup.bat

* + Test if it is ok

curl -H Accept:application/rdap+json http://$RDAP\_SERVER\_IP:$RDAP\_SERVER\_PORT/rdap/autnum/2100

It's ok if response contains 'rdapConformance'.

## rdap-proxy43

1. Get executable jar file 'rdap-proxy43-jar-with-dependencies.jar'.

There are two methods to get this file:

* + Get from [here](https://github.com/cnnic/rdap/raw/dev/rdap-proxy43/build/rdap-proxy43-jar-with-dependencies.jar) for jar build with JDK7.
  + [Build from source](https://github.com/cnnic/rdap/wiki/Proxy43-install:build-from-source)

1. Copy rdap-proxy43-jar-with-dependencies.jar to proxy43 install directory, and we call it $PROXY43\_INSTALL\_DIR.
2. Download configuration file "proxy43.properties" from [here](https://raw.githubusercontent.com/cnnic/rdap/dev/rdap-proxy43/src/main/resources/proxy43.properties), and copy it to $PROXY43\_INSTALL\_DIR.

You can edit this file for production use, see [here](https://github.com/cnnic/rdap/wiki/proxy43.properties)

1. Start up. **Must use root user**.
   * Start up

[in Linux/OS X, open a shell and execute command:]

cd $PROXY43\_INSTALL\_DIR #$PROXY43\_INSTALL\_DIR must be replaced by real dir

nohup java -jar rdap-proxy43-jar-with-dependencies.jar start &

[in Windows, open command prompt window and execute command:]

cd $PROXY43\_INSTALL\_DIR #$PROXY43\_INSTALL\_DIR must be replaced by real dir

java -jar rdap-proxy43-jar-with-dependencies.jar start

* + Test if it is ok

Run jwhois command, in Linux/OS X for example:

whois -h $PROXY43\_HOST cnnic.cn #$PROXY43\_HOST must be replaced by real proxy43 host

It's ok if response contains 'rdapConformance'.

* + Shutdown

cd $PROXY43\_INSTALL\_DIR #$PROXY43\_INSTALL\_DIR must be replaced by real dir

java -jar rdap-proxy43-jar-with-dependencies.jar shutdown

# Configuration



## Database configuration

$TOMCAT\_HOME/webapps/rdap/WEB-INF/classes/jdbc.properties

For test purpose, the first 4 properties - 'jdbc.url.hostPort','jdbc.url.dbName','jdbc.username','jdbc.password' must be configured.

(lines start with '#' are comments)

#jdbc url host and port.

#MUST change $MYSQL\_HOST\_OR\_IP to Mysql host or ip

jdbc.url.hostPort=jdbc:mysql://$MYSQL\_HOST\_OR\_IP:3306/

#database name

jdbc.url.dbName=rdap

#value change to jdbc username

jdbc.username=\*\*

#value change to jdbc password

jdbc.password=\*\*

#jdbc url params

jdbc.url.params=useUnicode=true&characterEncoding=UTF-8

# jdbc driver class name

jdbc.driverClassName=com.mysql.jdbc.Driver

#jdbc max pool size

jdbc.maxPoolSize=100

#jdbc min pool size

jdbc.minPoolSize=3

## Common configuration

$TOMCAT\_HOME/webapps/rdap/WEB-INF/classes/rdap.properties

For test purpose, the first 3 properties - 'localServiceUrl','inTlds','notInTlds' must be configured.

(lines start with '#' are comments)

#local service url,without scheme.This value is used in redirect service,

#to check if redirect url is local service url, and ignore the redirect

#if is local service url.

localServiceUrl=http://rdap.restfulwhois.org

#puny name of tlds in this registry, splited by ';'.Only in this list can

#be query.

inTlds=cn;xn--fiqs8s;arpa

#tlds not in this registry, splited by ';'.

#tlds in this list can not be query,and will query redirect instead.

#NOT-IN-TLDS has higher priority than IN-TLDs.

notInTlds=edu.cn

#max size for search.

maxsizeSearch=5

#batch size for search.

batchsizeSearch=100

#minSecondsAccessInterval for anonymous request. <=0 means no limit.

minSecondsAccessIntervalAnonymous=-1

#minSecondsAccessInterval for authenticated request. <=0 means no limit.

minSecondsAccessIntervalAuthed=-1

#max concurrent query count. 0 means no limit. Should less than web container's max threads.

maxConcurrentCount=0

#ipWhiteListForAccessInterval.proxy43'ip may put into this list.

ipWhiteListForAccessInterval=127.0.0.1;

#Requests from these IPs can be handled, and others will return 403 error.

ipWhiteListForUpdateApi=127.0.0.1;0:0:0:0:0:0:0:1

#not implemented uri,splited by ';'

#if this valid values not null ,it must be a combination of '/help' '/domains' '/domain/'

#'/entity/' '/entities' '/nameserver/' '/nameservers' '/autnum/' '/ip/' and splited by ';' .

# else the valid values is null

notImplementedUri=

#custom property prefix, NIC name is recommended.

customPropertyPrefix=cnnic\_

## Bootstrap configuration

$TOMCAT\_HOME/webapps/rdap/WEB-INF/classes/bootstrap.properties

This is used for bootstrap service in RDAP service, it will synchronize data from IANA periodically.

(lines start with '#' are comments)

#IANA bootstrap registry base URL

bootstrapRegistryBaseUrl=

#bootstrap URI for domain

bootstrapRegistryUriForDomain=domain.jsp

#bootstrap URI for as

bootstrapRegistryUriForAs=as.jsp

#bootstrap URI for ipv4

bootstrapRegistryUriForIpv4=ipv4.jsp

#bootstrap URI for ipv6

bootstrapRegistryUriForIpv6=ipv6.jsp

#CronExpression

cron.bootstrap=0 0 0 1 1 ?

(for CronExpression, please ref [Quartz cron expression](http://quartz-scheduler.org/api/2.2.0/org/quartz/CronExpression.html))

# Query API



## Introduction

* Only support HTTP 'GET' method
* Media type must be 'application/rdap+json' or ‘application/json’, in 'Accept' header.
* URI and parameters must be encoded in UTF-8.
* Unknown parameters will be ignored.
* Support HTTP BASIC authentication.When useing BASIC authentication, HTTPS must be used.
* Response is in JSON format. See 'Response Code' section for Response code.

Examples:

* Request without authentication:

Request URL: http://rdap.restfulwhois.org/entity/et-1

Request Method:GET

Accept:application/rdap+json

* Request with HTTP BASIC authentication:

Request URL: https://rdap.restfulwhois.org/entity/et-1

Request Method:GET

Accept:application/rdap+json

Authorization: BASIC $BASE64\_ENCODED\_USERNAME\_PASSWORD

($BASE64\_ENCODED\_USERNAME\_PASSWORD must be replaced by base64 encoded "username:password" string. Certificate for rdap.restfulwhois.org is [here](https://github.com/cnnic/rdap/raw/master/rdap-service/build/rdap.cer))

## Response Code

| **Response Code** | **When** |
| --- | --- |
| 200 | Ok |
| 404 | Not found for query |
| 400 | Request URI is invalid, or parameter is invalid |
| 422 | Unsupported search string : more than one '\*' in q, or q starts with '\*', or q is '\*' |
| 415 | Request with invalid media type: 'Accept' header is not 'application/rdap+json' |
| 401 | Unauthorized: authentication failed |
| 403 | Forbidden: the query object is forbidden for this client |
| 405 | Method Not Allowed. Only 'GET' method is allowed |
| 500 | Internal Server Error |
| 301 | Moved Permanently, client should request for the url by 'Location' in Response header |
| 429 | Too Many Requests, client should wait for a moment before query |
| 509 | Bandwidth Limit Exceeded: server is busy, client should try later |

## Api

#### Query Entity

URI: /entity/<handle>

Example: http://rdap.restfulwhois.org/entity/et-1

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-entity-query) for the response body.

#### Query Nameserver

URI: /nameserver/<name server name>

Example: http://rdap.restfulwhois.org/nameserver/xn--1-dr6av31f.xn--0zwm56d.xn--fiqs8s

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-nameserver-query) for the response body.

#### Query Domain

URI: /domain/<domain name>

Example: http://rdap.restfulwhois.org/domain/cnnic.cn

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-domain-query) for the response body.

#### Query IP Network

URI: /ip/<IP address> or ip/<CIDR prefix>/<CIDR length>

Example:

http://rdap.restfulwhois.org/ip/218.0.0.3

http://rdap.restfulwhois.org/ip/218.241.0.0/30

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-IP-network-query) for the response body.

#### Query Autonomous System Number

URI: /autnum/<autonomous system number>

Example: http://rdap.restfulwhois.org/autnum/1

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-autnum-query) for the response body.

#### Query Help

URI: /help

Example: http://rdap.restfulwhois.org/help

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-help) for the response body.

#### Search Domain

URI:

/domains?name=<domain search pattern>

/domains?nsLdhName=<ns ldhName search pattern>

/domains?nsIp=<ns IP>

Example:

http://rdap.restfulwhois.org/domains?name=c\*

http://rdap.restfulwhois.org/domains?nsLdhName=ns1.host\*.cn

http://rdap.restfulwhois.org/domains?nsIp=218.241.111.96

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-domain-search) for the response body.

#### Search Name Server

URI: /nameservers?name=<nameserver search pattern>

URI: /nameservers?ip=<IP Network search pattern>

Example:

http://rdap.restfulwhois.org/nameservers?name=n\*cn

http://rdap.restfulwhois.org/nameservers?ip=218.241.111.96

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-nameserver-search) for the response body.

#### Search Entity

URI: /entities?fn=<entity name search pattern>

URI: /entities?handle=<entity handle search pattern>

Example:

http://rdap.restfulwhois.org/entities?fn=Jo\*n

http://rdap.restfulwhois.org/entities?handle=et\*

[Click here](https://github.com/cnnic/rdap/wiki/The-response-of-the-entity-search) for the response body.

# Update API

The RDAP data can be updated by update API.



## Introduction

* All Update API prefix: /u/
* Content type must be 'application/rdap+json' or 'application/json', in 'Content-Type' header.
* URI and parameters must be encoded in UTF-8.
* Unknown parameters will be ignored.
* Security consideration: Update API Support IP authentication, to only allow request which IP is in white list.
* Request and Response body is in JSON format.
* About 'handle': chars should be in [ASCII - \_].
* Max length of columns: for 'handle' value is 100, all others are 255 if not specified in the following tables.

## Common request format

### create

* HTTP METHOD : POST
* URI : /u/{objectType}  
  objectType：domain、nameserver、ip、autnum、entity
* CONTENT TYPE : 'application/rdap+json' or 'application/json'
* BODY: JSON formatted key-value parameters.

### update

* HTTP METHOD : PUT
* URI : /u/{objectType}/{handle}  
  handle：object handle
* CONTENT TYPE : 'application/rdap+json' or 'application/json'
* BODY : the same with 'create'

### delete

* HTTP METHOD : DELETE
* URI : /u/{objectType}/{handle}  
  handle：object handle

## Common response format

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP status code** | **service code** | **body** | **description** |
| 200 |  | {"handle":"xxx"} | success response |
| not 200 |  | {"handle":"domain-1",  "errorCode":400,  "subErrorCode":4002,  "description":[" Property can’t be empty:ldhName"]} | failure response |

## Response code

|  |  |  |
| --- | --- | --- |
| **HTTP status code** | **service code** | **description** |
| 200 |  | success response |
| 400 | 4001 | Request data is not valid JSON, or has invalid date type |
| 400 | 4002 | Property can't be empty |
| 400 | 4003 | Property exceed max length |
| 400 | 4007 | Property must be valid date |
| 400 | 4008 | Property value is not valid |
| 400 | 4009 | Unrecognized request URI |
| 400 | 40010 | Property value must between [start, end] |
| 403 | 4031 | Forbidden |
| 404 | 4041 | Object not found with handle |
| 409 | 4091 | Object already exist for handle |
| 405 |  | method not allowed |
| 415 |  | unsupported media type |
| 500 |  | internal server error |

## Request body parameter

Request body parameter, for CREATE and UPDATE Request.

### Common parameter

All update API can have these parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **type** | **length/range** | **not empty** | **description** |
| handle | string | 1-100 | Y | registry-unique identifiers referenced an object. Should be ASCII and '-'/'\_'. |
| entities | array |  | N | arrays of inner-entity object |
| status | array | 0-20 | N | status array, each status length must be [0-20]. e.g: [“validated”,”redacted”] |
| remarks | array |  | N | arrays of remark object |
| links | array |  | N | arrays of link object |
| events | array |  | N | arrays of event object |
| lang | string | 0-64 | N | Language Identifier, e.g: "en" |
| port43 | string | 0-4096 | N | Port 43 WHOIS Server |
| customProperties | object |  | N | e.g: {"customKey1":"value1","customKey2":"value2" } |

#### inner-object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **type** | **length/range** | **not empty** | **description** |
| handle | string | 1-100 | Y | object handle. Non-exist handle will be ignored. |

#### inner-entity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **type** | **length/range** | **not empty** | **description** |
| handle | string | 1-100 | Y | entity handle.  Non-exist handle will be ignored. |
| roles | array |  | N | e.g: [“registrant”, “administror”] |

#### remark or notice

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| title | string | N | title of the object |
| description | array | N | each description length must be [0-2048] |
| links | array | N | arrays of link object |

#### link

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| value | string | N | [0-2048]. e.g: "<http://example.com/context_uri>" |
| rel | string | N | e.g: "self" |
| href | string | N | e.g: "<http://example.com/target_uri>" |
| hreflang | array | N | e.g: [ "en", "ch" ] |
| title | string | N | <http://tools.ietf.org/html/rfc5988#section-5> |
| media | string | N | e.g: "screen" |
| type | string | N | e.g: "application/json" |

#### publicId

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| type | string | N | a string denoting the type of public identifier |
| identifier | string | N | a public identifier of the type denoted by 'type' |

#### event

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| eventAction | string | Y | a string denoting the reason for the event |
| eventActor | string | N | denoting the actor responsible for the event |
| eventDate | string | Y | UTC date time. Format must be like: 2015-01-01T01:01:01Z |
| links | array | N | arrays of link object |

### Domain

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| ldhName | string | Y | puny name of domain. Can't contain last '.' of domain. Must be lowercased. |
| unicodeName | string | N | [0,1024]. unicode name of domain. If is ASCII domain then it is the same with ldhName |
| variants | array | N | array of variant object |
| nameservers | array | N | arrays of inner-object object |
| secureDNS | object | N | secureDNS object |
| publicIds | array | N | arrays of publicId, e.g: [{"type":"IANA Registrar ID", "identifier":"1"}] |
| type | string | N | "dnr" for DNR domain, or "arpa" for ARPA domain |
| networkHandle | string | N | network handle for ARPA domain. This value will be ignored if network not exist. |

#### variant

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| relation | array | N | array of relation string. e.g: [ "registered", "conjoined" ] |
| idnTable | string | N | the name of the IDN table of codepoints |
| variantNames | array | N | array of variantName object |

#### variantName

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| ldhName | string | N | variant's ldhName.Can't contain last '.' of domain. Must be lowercased. |
| unicodeName | string | N | variant's unicodeName |

#### secureDNS

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| zoneSigned | boolean | N | true if the zone has been signed, false otherwise. |
| delegationSigned | boolean | N | boolean true if there are DS records in the parent, false otherwise. |
| maxSigLife | int | N | the signature life time in seconds to be used when creating the RRSIG DS record |
| dsData | array | N | array of dsData object |
| keyData | array | N | array of keyData object |

#### dsData

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| keyTag | int | Y | the key tag field of a DNS DS record as specified by RFC 4034 |
| algorithm | int | Y | the algorithm field of a DNS DS record as described by RFC 4034 |
| digest | string | Y | [0-2048]. the digest field of a DNS DS record as specified by RFC 4034 |
| digestType | int | Y | the digest type field of a DNS DS record as specified by RFC 4034 |
| links | array | N | arrays of link object |
| events | array | N | arrays of event object |

#### keyData

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| flags | int | Y | the flags field value in the DNSKEY record as specified by RFC 4034 |
| protocol | int | Y | the protocol field value of the DNSKEY record as specified by RFC 4034 |
| publicKey | string | N | the public key in the DNSKEY record as specified by RFC 4034 |
| algorithm | int | Y | the algorithm field of a DNSKEY record as specified by RFC 4034 |
| links | array | N | arrays of link object |
| events | array | N | arrays of event object |

### Nameserver

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| ldhName | string | Y | puny name of nameserver. Can't contain last '.' of domain. Must be lowercased. |
| unicodeName | string | N | [0,1024]. If is ASCII nameserver then it is the same with ldhName |
| ipAddresses | object | N | ipAddresses object |

#### ipAddresses

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| ipList | array | N | arrays of IP. IP can be v4 or v6. e.g: [“218.1.1.1”, “2001:db8::”] |

### Entity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **type** | **length/range** | **not empty** | **description** |
| fn | string |  | Y | entity name |
| kind | string |  | N | <http://tools.ietf.org/html/rfc6350#section-6.1.4> |
| email | string |  | N | email |
| title | string |  | N | <http://tools.ietf.org/html/rfc6350#section-6.6.1> |
| org | string |  | N | org |
| url | string | 0-4096 | N | <http://tools.ietf.org/html/rfc6350#section-6.7.8> |
| addresses | array |  | N | array of address object |
| telephones | array |  | N | array of telephone telephones |
| publicIds | array |  | N | the same with domain |

#### address

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| pref | string | N | <http://tools.ietf.org/html/rfc6350#section-5.3> |
| types | string | N | multiple type string separated by ';'.<http://tools.ietf.org/html/rfc6350#section-5.6> |
| postbox | string | N | postbox |
| extendedAddress | string | N | the extended address |
| streetAddress | string | N | street address |
| locality | string | N | the locality. e.g: city |
| region | string | N | the region. e.g:state or province |
| postalcode | string | N | the postal code |
| country | string | N | the country name |

#### telephone

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| pref | string | N | <http://tools.ietf.org/html/rfc6350#section-5.3> |
| types | string | N | string of type for multiple<http://tools.ietf.org/html/rfc6350#section-5.6>, separated by';' |
| number | string | Y | telephone number |
| extNumber | string | N | extended number |

### Network

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| startAddress | string | Y | the starting number in the block of network |
| endAddress | string | Y | the ending number in the block of network |
| ipVersion | string | N | 'v4' or 'v6'. This value will not affect the real type for startAddress and endAddress. |
| name | string | N | an identifier assigned to the network registration by the registration holder |
| type | string | N | a string containing an RIR-specific classification of the network |
| country | string | N | a string containing the two-character country code of the network |
| parentHandle | string | N | parent network of this network registration |
| cidr | string | Y | formatted network used to generate self link for query. <http://tools.ietf.org/html/rfc4632>. e.g: 92.168.99.0/24 |

### As number

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **not empty** | **description** |
| startAutnum | string | Y | the starting number in the block of autonomous system numbers |
| endAddress | string | Y | the ending number in the block of autonomous system numbers |
| name | string | N | an identifier assigned to the autnum registration by the registration holder |
| type | string | N | a string containing an RIR-specific classification of the autnum |
| country | string | N | a string containing the name of the 2 character country code of the autnum |

# Proxy43



## Introduction

* Only support jwhois client
* Response is JSON formatted

Usage:

whois [type] [parameter]

type: query type. Values are: ""(empty), nameserver, entity, as, domains, nameservers, entities.

parameter: query parameter.

## Api

#### Query IP

whois {IP}

type: empty

parameter: IP address

examples:

whois 218.241.111.44

whois 218.241.111.44/8

whois 3000:0DB8:0000:0000:0000:0000:1428:0000/128

#### Query domain

whois {domain name}

type: empty

parameter: domain name, not IP formatted

examples:

whois cnnic.cn

#### Search domain by domain name

whois domains {domain name search pattern}

type: domains

parameter: domain name search pattern

examples:

whois domains cnnic\*.cn

#### Search domain by domain's nameserver name

whois domains nsLdhName={nameserver name search pattern}

type: domains

parameter: nameserver name search pattern

examples:

whois domains nsLdhName=ns.cnnic\*.cn

#### Search domain by IP of domain's nameserver

whois domains nsIp={IP}

type: domains

parameter: IP of domain's nameserver

examples:

whois domains nsIp=218.241.111.44

#### Query nameserver

whois nameserver {nameserver name}

type: nameserver

parameter: nameserver name

examples:

whois nameserver ns.cnnic.cn

#### Search nameserver by name

whois nameservers {nameserver name search pattern}

type: nameservers

parameter: nameserver name search pattern

examples:

whois nameservers ns\*.cn

#### Search nameserver by IP of nameserver

whois nameservers ip={IP of nameserver}

type: nameservers

parameter: IP of nameserver

examples:

whois nameservers ip=218.241.111.44

#### Query as number

whois as {as number}

type: as

parameter: as number

examples:

whois as 2345

#### Query entity

whois entity {entity handle}

type: entity

parameter: entity handle

examples:

whois entity handle\_of\_an\_entity

#### Search entity by entity name

whois entities fn={entity name search pattern}

type: entities

parameter: entity name search pattern

examples:

whois entities fn=John\*

#### Search entity by entity handle

whois entities fn={entity handle search pattern}

type: entities

parameter: entity handle search pattern

examples:

whois entities handle=handle\_of\_John\*

# Customize and Develop



## Use Registry’s database

Registry can modify code to use its own database and schema, instead of updating date from API periodically.

Steps:

1. If database is not MYSQL, you should change database driver:
   * Modify pom.xml, remove mysql dependency. Then add your database dependency.
   * Modify [database configuration](https://github.com/cnnic/rdap/wiki/jdbc.properties)
2. Modify JAVA code of DAO implementation.

Modify all DAO implementation JAVA class, such as DomainQueryDaoImpl:

* + Change DAO implementation according to your own database schema.
  + If you need more properties than default, you can [add your custom properties](https://github.com/cnnic/rdap/wiki/add-custom-properties-values) .

## Function pick

RDAP server supports all 6 query functions and 3 search functions defined in draft-ietf-weirds-rdap-query-15.

* IP network query
* autonomous system number query
* domain query
* nameserver query
* entity query
* help query
* domain search
* nameserver search
* entity search

You can disable some of these functions by adding the function URI to ‘notImplementedUri’ property in [rdap.properties](https://github.com/cnnic/rdap/wiki/rdap.properties).

## Customize Validator

Query/search parameter are validated before query.

You can modify validation logic by add/remove/modify validators.

All validators extend from [Validator.java](https://github.com/cnnic/rdap/tree/master/rdap-webapp/src/main/java/org/restfulwhois/rdap/core/common/validation), such as [DomainNameValidator](https://github.com/cnnic/rdap/blob/master/rdap-service/src/main/java/org/restfulwhois/rdap/core/domain/validator/DomainNameValidator.java).

## Enable/disable access control

Access Control is done in AccessControlQueryFilter.  
You can enable/disable access control by configuring 'accessControlQueryFilter' in [queryFilter](https://github.com/cnnic/rdap/wiki/query%20filter).

## Enable/disable redirect

Redirect is done in \*RedirectQueryFilter.

You can enable/disable redirect by configuring these filters in [queryFilter](https://github.com/cnnic/rdap/wiki/query%20filter).

## Add custom features

You can add custom features by adding [queryFilter](https://github.com/cnnic/rdap/wiki/query%20filter).

## VCARD extension

We use [Jcard](https://github.com/cnnic/rdap/blob/dev/rdap-service/src/main/java/org/restfulwhois/rdap/core/entity/model/jcard/Jcard.java) to convert VCARD to JSON.  
Jcard convert property by property, using [JcardPropertyConverter](https://github.com/cnnic/rdap/blob/master/rdap-service/src/main/java/org/restfulwhois/rdap/core/entity/model/jcard/JcardPropertyConverter.java).  
If you need to show more vcard information, you can add your custom converter:

* add your own implementation, extending JcardPropertyConverter.
* registrar this class to Jcard by adding it to converters in Jcard.java.

# Other

Other information can be found in project wiki: https://github.com/cnnic/rdap/wiki.