Cloud DNS

Developer Guide

API v1.0 BETA (Mar. 21, 2011)

rackspace® HOSTING

docs.rackspacecloud.com/api

Cloud DNS Developer Guide

API v1.0 BETA (2011-03-21)

This document is intended for software developers interested in developing applications using the Rackspace Cloud DNS Application Programming Interface (API). The document is for informational purposes only and is provided "AS IS."

Table of Contents

List of Tables

List of Examples

1. Overview

Rackspace Cloud DNS is a Domain Name System (DNS) available to Rackspace Cloud customers. Interactions with Rackspace Cloud DNS occur programmatically via the Rackspace Cloud DNS API as described in this Cloud DNS Developer Guide.

We welcome feedback, comments, and bug reports at support@rackspacecloud.com.

1.1. Intended Audience

This Guide is intended to assist software developers who want to develop applications using the DNS Service API. To use the information provided here, you should first have a general understanding of the DNS service. You should also be familiar with:

- DNS terminology
- general operating principles of DNS
- ReSTful web services
- HTTP/1.1 conventions
- JSON and/or XML data serialization formats
- Atom syndication format

1.2. Document Change History

This version of the Developer Guide replaces and obsoletes all previous versions. The most recent changes are described in the table below:

Revision Date	Summary of Changes
Mar. 21, 2011	• Initial release for Private Beta.

1.3. Additional Resources

You can download the most current versions of other API-related documents from http://docs.rackspacecloud.com/api/.

For information about Rackspace Cloud products, refer to http://www.rackspace.com/cloud. This site also offers links to Rackspace's official support channels, including knowledge base articles, forums, phone, chat, and email.

You can also follow Rackspace updates and announcements via twitter at http://www.twitter.com/rackcloud .

This API uses standard HTTP 1.1 response codes as documented at http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html .

2. Concepts

To use the DNS API effectively, you should understand several key concepts:

2.1. DNS

The Domain Name System (DNS) is a system by which internet domain name-to-address and address-to-name resolutions are determined. All domains and their components, such as mail servers, utilize DNS to resolve to the appropriate locations. DNS servers are usually set up in a master-slave relationship such that failure of the master invokes the slave. DNS servers may also be clustered or replicated such that changes made to one DNS server are automatically propagated to other active servers.

Note

DNS understands only ASCII, so the Cloud DNS Service provides conversion between UTF-8 and ASCII on all calls into the system.

2.2. Domain

A domain is an entity/container of all DNS-related information containing one or more records.

2.3. Subdomain

Subdomains are domains within a parent domain. Subdomains can themselves have subdomains, so third-level, fourth-level, fifth-level, and deeper levels of nesting are possible.

2.4. Record

A DNS record belongs to a particular domain and is used to specify information about the Domain. There are several types of DNS Records; each record type contains particular information used to describe that record's purpose. Examples include Mail Exchange (MX) records which specify the mail server for a particular domain and Name Server (NS) records which specify name servers for a domain.

2.5. Domain Owner

The domain owner is the entity that owns the domain information. The domain owner is not necessarily the same entity that owns any or all subdomains within the domain. A domain owner may delegate ownership of subdomains or the parent domain to another entity.

3. General API Information

The DNS Service API is implemented using a ReSTful web service interface. Like other products in the Rackspace Cloud suite, the DNS Service shares a common token-based authentication system that allows seamless access between products and services.

Note

All requests to authenticate against and operate the service are performed using SSL over HTTP (HTTPS) on TCP port 443.

3.1. Authentication

Each ReST request against the DNS service requires the inclusion of a specific authorization token HTTP x-header, defined as `X-Auth-Token`. Customers obtain this token by first using the Rackspace Cloud Authentication Service and supplying a valid username and API access key.

The Rackspace Cloud Authentication Service is a ReSTful web service. It is the entry point to all Rackspace Cloud APIs. It is accessible at https://auth.api.rackspacecloud.com/v1.0.

Request

To authenticate, you must supply your username and API access key in x-headers:

- Use your Rackspace Cloud username as the username for the API. Place it in the `X-Auth-User` x-header.
- Obtain your API access key from the Rackspace Cloud Control Panel in the Your Account | API Access section. Place it in the `X-Auth-Key` x-header.

Example 3.1. Authentication Request

```
GET /v1.0 HTTP/1.1
Host: auth.api.rackspacecloud.com
X-Auth-User: jdoe
X-Auth-Key: a86850deb2742ec3cb41518e26aa2d89
```

Response

If authentication is successful, an HTTP status 204 (No Content) is returned with an `X-Auth-Token` header; additional Cloud Service headers are returned along with with `X-Auth-Token`, but they are not applicable to the DNS service. An HTTP status of 401 (Unauthorized) is returned if authentication fails. All operations against the DNS service must include the `X-Auth-Token` header as noted above.

Example 3.2. Authentication Response

```
HTTP/1.1 204 No Content
Date: Mon, 12 Nov 2007 15:32:21 GMT
Server: Apache
X-Server-Management-Url: https://servers.api.rackspacecloud.com/
v1.0/35428
X-Storage-Url: https://storage.clouddrive.com/v1/
CloudFS_9c83b-5ed4
X-CDN-Management-Url: https://cdn.clouddrive.com/v1/
CloudFS_9c83b-5ed4
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

Authentication tokens are typically valid for 24 hours. Applications should be designed to re-authenticate after receiving a 401 (Unauthorized) response.

3.2. Service Access

The DNS service is a global service which allows for a caller to have a single endpoint against which all service operations are accessible. DNS is therefore responsible for appropriate replication, caching, and overall maintenance of DNS data across regional boundaries to other DNS servers.

To access the global endpoint, replace the sample account ID number, *1234*, with your actual account number as returned in the authentication service response.

Table 3.1. Global Service Endpoint

Global Endpoint	Endpoint Location
https://ord.dns.api.rackspacecloud.com/v1.0/*1234*	Chicago (ORD)

3.3. Request/Response Types

The DNS API supports both the JSON and XML data serialization formats. The request format is specified using the `Content-Type` header and is required for operations that have a request body. The response format can be specified in requests either by using the `Accept` header or by adding an `.xml` or `.json` extension to the request URI. Note that it is possible for a response to be serialized using a format different from the request. If no response format is specified, JSON is the default. If conflicting formats are specified using both an `Accept` header and a query extension, the query extension takes precedence.

Some operations support an Atom representation that can be used to efficiently determine when the state of services has changed.

Table 3.2. Response Formats

Format	Accept Header	Query Extension	Default
JSON	application/json	.json	Yes
XML	application/xml	.xml	No
Atom	application/atom+xml	.atom	No

3.4. Content Compression

Request and response body data may be encoded with gzip compression to accelerate interactive performance of API calls and responses. This is controlled using the `Accept-Encoding` header on the request from the client and indicated by the `Content-Encoding` header in the server response. Unless the header is explicitly set, encoding defaults to disabled.

Table 3.3. Encoding Headers

Header Type	Name	Value
HTTP/1.1 Request	`Accept-Encoding`	gzip
HTTP/1.1 Response	`Content-Encoding`	gzip

3.5. Persistent Connections

By default, the API supports persistent connections via HTTP/1.1 keepalives. All connections will be kept alive unless the connection header is set to close.

To prevent abuse, HTTP sessions have a timeout of 20 seconds before being closed.

Note

The server may close the connection at any time and clients should not rely on this behavior.

3.6. Limits

All accounts, by default, have a preconfigured set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: *rate limits* and *absolute limits*. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed.

Note

If the default limits are too low for your particular application, please contact Rackspace Cloud support to request an increase. All requests require reasonable justification.

3.6.1. Rate Limits

We specify rate limits in terms of both a human-readable wild-card URI and a machine-processable regular expression. The regular expression boundary matcher '^' takes effect after the root URI path. For example, the regular expression ^/v1.0/*1234*/domains would match the bolded portion of the following URI: https://ord.dns.api.rackspacecloud.com/**v1.0/1234/domains**.

Table 3.4. Default Rate Limits

Verb	URI	RegEx	Default
GET	/v1.0/*	^/v1.0/.*	5/second
GET	/v1.0/*	^/v1.0/.*	100/minute
POST	/v1.0/*	^/v1.0/.*	2/second
POST	/v1.0/*	^/v1.0/.*	25/minute
PUT	/v1.0/*	^/v1.0/.*	5/second
PUT	/v1.0/*	^/v1.0/.*	50/minute
DELETE	/v1.0/*	^/v1.0/.*	2/second
DELETE	/v1.0/*	^/v1.0/.*	50/second

Rate limits are applied in order relative to the verb, going from least to most specific. For example, although the threshold for **POST** to /v1.0/* is 25 per minute, one cannot **POST** to /v1.0/* more than 2 times within a single second because the rate limit for any **POST** is 2 per second.

If you exceed the thresholds established for your account, a 413 (Rate Control) HTTP response will be returned with a `Reply-After` header to notify the client when they can attempt to try again.

3.6.2. Absolute Limits

POST and **PUT** calls are limited to the creation or modification of a maximum of 100 entities per call where an entity is defined as a record, domain, or subdomain. For example, when using **POST** `/domains` to create a new domain with nine subdomains, you could create a maximum of ninety records across the domain and subdomains. This would total 100 entitities: 1 domain + 9 subdomains + 90 records. Additional records and/or subdomains could be created for the domain in subsequent calls.

4. API Operations

4.1. Domains

In the following examples, *1234* should be replaced by your authenticated account ID.

Verb	URI	Description	Example Call
GET	/version	List version information for specific components of the API service.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/version
GET	/domains	List all account domains.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains
GET	/domains/detail	List all account domains and their detailed information.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains/detail
GET	/domains/*domainId*	List details for a specific domain.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains/*765437*
POST	/domains	Create a new domain.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains
PUT	/domains*domainId*	Modify the configuration of a domain.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains/*765437*
PUT	/domains	Modify multiple domains.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains
DELETE	/domains*domainId*	Remove a domain.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains/*765437*
DELETE	/domains*domainId*? deleteSubdomains=true	Remove a domain and all its subdomains.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains/*765437*? deleteSubdomains=true
DELETE	/domains?id=*domainId1*& id=*domainId2*	Remove multiple domains.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains? id=*765437*&id=*765438*
DELETE	/domains?id=*domainId1*& id=*domainId2*& deleteSubdomains=true	Remove multiple domains and their subdomains.	https://ord.dns.api.rackspacecloud.com/v1.0/*1234*/domains? id=*765437*&id=*765438*& deleteSubdomains=true

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

4.1.1. List Domains

Verb	URI	Description	Representations
GET	/domains	List all domains and subdomains manageable by the account specified. Display IDs and names only.	XML, JSON
GET	/domains/detail	List all domains and subdomains manageable by the account specified. Display all details, including records.	XML, JSON

Normal Response Code(s): 200

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

These operations provide a list of all DNS domains and subdomains manageable by a given account. The type of display information is controlled by whether or not detailed information has been requested. Filtering can be accomplished by specifying additional parameters. For example, `?name=hoola.com` matches `hoola.com` and similar names such as `main.hoola.com` and `sub.hoola.com`.

Note

The wildcard is assumed from the left but not the right.

Example 4.1. List Domains Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/
```

Example 4.2. List Domains Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<domains xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
 xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <domain emailAddress="email@bbunitteststatic3.com" ttl="5400"
 name="bbunitteststatic3.com" id="2610217">
        <nameservers>
            <nameserver name="ns.rackspace.com"/>
            <nameserver name="ns1.rackspace.com"/>
        </nameservers>
        <records>
            <record ttl="5400" data="ns1.rackspace.com
 email@bbunitteststatic3.com 5403 5404  5401 5402 1297891127"
                name="bbunitteststatic3.com" type="SOA"
 id="2610217"/>
            <record ttl="5000" data="ns2.racKspAce.com"
 name="bbunitteststatic3.com" type="NS" id="6012416"/>
            <record priority="1" ttl="5000"
 data="mail.bbunitteststatic3.com" name="bbunitteststatic3.com"
 type="MX" id="3012531"/>
            <record ttl="5000" data="111.111.111.111"
 name="singleA.bbunitteststatic3.com" type="A" id="6500179"/>
            <record ttl="5000"
 data="2001:db8:1f70:0:999:de8:7648:6e8"
 name="quadA.bbunitteststatic3.com" type="AAAA" id="94"/>
            <record ttl="5000" data="Default text vAlue"
 name="bbunitteststatic3.com" type="TXT" id="226409"/>
            <record ttl="5000" data="foo.bbunitteststatic3.com"
 name="fqdn.bbunitteststatic3.com" type="CNAME" id="9573348"/>
            <record ttl="5000" data="sub.bbunitteststatic3.com"
 name="pub.bbunitteststatic3.com" type="DNAME" id="10"/>
            <record priority="1" ttl="5000"
 data="sipserver.bbunitteststatic3.com"
 name="sip.bbunitteststatic3.com" type="SRV" id="7318"/>
        </records>
    </domain>
</domains>
```

Example 4.3. List Domains Response: JSON

```
{
  "domains" : {
    "domain" : [ {
      "id" : 2610338,
      "name" : "bbunittestacct2static3.com"
    }, {
      "id" : 2610368,
      "name" : "demodomainwithsubs.com",
      "subdomains" : {
        "domain" : [ {
          "id" : 2610369,
          "name" : "sub.demodomainwithsubs.com"
        } ]
      }
    }, {
      "id" : 2610326,
      "name" : "domainremoverecord.com"
    }, {
      "id" : 2610424,
      "name" : "demo.dayjson.org",
      "subdomains" : {
        "domain" : [ {
          "id" : 2610425,
          "name" : "sub.demo.dayjson.org"
        } ]
      }
    }, {
      "id" : 2575497,
      "name" : "cwc201101201654.com"
    }]
  }
}
```

Example 4.4. List Domains Detail Request: XML

```
http://localhost/cxf/dns/v1.0/440370/domains/detail
```

Example 4.5. List Domains Detail Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<domains xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
 xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <domain emailAddress="email@bbunitteststatic3.com" ttl="5400"
 name="bbunitteststatic3.com" id="2610217">
        <nameservers>
            <nameserver name="ns.rackspace.com"/>
            <nameserver name="ns1.rackspace.com"/>
        </nameservers>
        <records>
            <record ttl="5400" data="ns1.rackspace.com
 email@bbunitteststatic3.com 5403 5404  5401 5402 1297891127"
                name="bbunitteststatic3.com" type="SOA"
 id="2610217"/>
            <record ttl="5000" data="ns2.racKspAce.com"
 name="bbunitteststatic3.com" type="NS" id="6012416"/>
            <record priority="1" ttl="5000"
 data="mail.bbunitteststatic3.com" name="bbunitteststatic3.com"
 type="MX" id="3012531"/>
            <record ttl="5000" data="111.111.111.111"
 name="singleA.bbunitteststatic3.com" type="A" id="6500179"/>
            <record ttl="5000"
 data="2001:db8:1f70:0:999:de8:7648:6e8"
 name="quadA.bbunitteststatic3.com" type="AAAA" id="94"/>
            <record ttl="5000" data="Default text vAlue"
 name="bbunitteststatic3.com" type="TXT" id="226409"/>
            <record ttl="5000" data="foo.bbunitteststatic3.com"
 name="fqdn.bbunitteststatic3.com" type="CNAME" id="9573348"/>
            <record ttl="5000" data="sub.bbunitteststatic3.com"
 name="pub.bbunitteststatic3.com" type="DNAME" id="10"/>
            <record priority="1" ttl="5000"
 data="sipserver.bbunitteststatic3.com"
 name="sip.bbunitteststatic3.com" type="SRV" id="7318"/>
        </records>
    </domain>
</domains>
```

Example 4.6. List Domains Detail Response: JSON

```
{
   "domains":{
      "domain":{
         "emailAddress":"email@bbunitteststatic3.com",
         "ttl":"5400",
         "name":"bbunitteststatic3.com",
         "id":"2610217",
         "nameservers":{
            "nameserver":[{
               "name":"ns.rackspace.com"
```

```
            },{
               "name":"ns1.rackspace.com"
            }]
         },
         "records":{
            "record":[{
               "ttl":"5400",
               "data":"ns1.rackspace.com
email@bbunitteststatic3.com 5403 5404  5401 5402 1297891127",
               "name":"bbunitteststatic3.com",
               "type":"SOA",
               "id":"2610217"
            },{
               "ttl":"5000",
               "data":"ns2.racKspAce.com",
               "name":"bbunitteststatic3.com",
               "type":"NS",
               "id":"6012416"
            },{
               "priority":"1",
               "ttl":"5000",
               "data":"mail.bbunitteststatic3.com",
               "name":"bbunitteststatic3.com",
               "type":"MX",
               "id":"3012531"
            },{
               "ttl":"5000",
               "data":"111.111.111.111",
               "name":"singleA.bbunitteststatic3.com",
               "type":"A",
               "id":"6500179"
            },{
               "ttl":"5000",
               "data":"2001:db8:1f70:0:999:de8:7648:6e8",
               "name":"quadA.bbunitteststatic3.com",
               "type":"AAAA",
               "id":"94"
            },{
               "ttl":"5000",
               "data":"Default text value",
               "name":"bbunitteststatic3.com",
               "type":"TXT",
               "id":"226409"
            },{
               "ttl":"5000",
               "data":"foo.bbunitteststatic3.com",
               "name":"fqdn.bbunitteststatic3.com",
               "type":"CNAME",
               "id":"9573348"
            },{
               "ttl":"5000",
```

```
                "data":"sub.bbunitteststatic3.com",
                "name":"pub.bbunitteststatic3.com",
                "type":"DNAME",
                "id":"10"
             },{
               "priority":"1",
               "ttl":"5000",
               "data":"sipserver.bbunitteststatic3.com",
               "name":"sip.bbunitteststatic3.com",
               "type":"SRV",
               "id":"7318"
             }]
          }
       }
    }
}
```

4.1.2. List Domain Details

Verb	URI	Description	Representations
GET	/domains/*domainId*	List details of the specified domain. Display all details, including records.	XML, JSON

Normal Response Code(s): 200

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation provides the detailed output for a specific domain configured and associated with an account. This operation is not capable of returning details for a domain which has been deleted.

This operation does not require a request body.

Example 4.7. List Domain Details Request: XML

```
http://localhost/cxf/dns/v1.0/434705/domains/2610368/
```

Example 4.8. List Domain Details Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<domains xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
    xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <domain emailAddress="demo@enduser.com" ttl="5600"
        name="demodomainwithsubs.com" id="2610368">
        <nameservers>
            <nameserver name="ns.rackspace.com" />
            <nameserver name="ns1.rackspace.com" />
        </nameservers>
        <records>
            <record ttl="5600"
                data="ns1.rackspace.com demo@enduser.com 3600 300
 3600 3600 1298349924"
                name="demodomainwithsubs.com" type="SOA"
 id="2610368" />
            <record ttl="5600" data="dns1.stabletransit.com"
 name="demodomainwithsubs.com"
                type="NS" id="6012767" />
            <record ttl="5600" data="dns2.stabletransit.com"
 name="demodomainwithsubs.com"
                type="NS" id="6012768" />
            <record ttl="86400" data="ns1.rackspace.com"
 name="demoDomainWithSubs.com"
                type="NS" id="6012769" />
            <record ttl="86400" data="ns2.rackspace.com"
 name="demoDomainWithSubs.com"
                type="NS" id="6012770" />
```

```
            <record priority="2" ttl="5000"
 data="mail.demoDomainWithSubs.com"
                name="demoDomainWithSubs.com" type="MX"
 id="3012556" />
            <record ttl="86400" data="172.11.122.24"
 name="demoDomainWithSubs.com"
                type="A" id="6500237" />
        </records>
        <subdomains>
            <domain emailAddress="hostmaster@secondlevel.com"
 ttl="5600"
                name="sub.demodomainwithsubs.com" id="2610369">
                <nameservers>
                    <nameserver name="ns.rackspace.com" />
                    <nameserver name="ns1.rackspace.com" />
                </nameservers>
                <records>
                    <record ttl="5600"
                        data="ns1.rackspace.com
 hostmaster@secondlevel.com 3600 300  3600 3600 1298349928"
                        name="sub.demodomainwithsubs.com"
 type="SOA" id="2610369" />
                    <record ttl="5600"
 data="dns1.stabletransit.com" name="sub.demodomainwithsubs.com"
                        type="NS" id="6012771" />
                    <record ttl="5600"
 data="dns2.stabletransit.com" name="sub.demodomainwithsubs.com"
                        type="NS" id="6012772" />
                    <record ttl="86400" data="ns1.rackspace.com"
 name="sub.demoDomainWithSubs.com"
                        type="NS" id="6012773" />
                    <record ttl="86400" data="ns2.rackspace.com"
 name="sub.demoDomainWithSubs.com"
                        type="NS" id="6012774" />
                    <record ttl="86400" data="172.11.122.23"
 name="sub.demoDomainWithSubs.com"
                        type="A" id="6500238" />
                </records>
            </domain>
        </subdomains>
    </domain>
</domains>
```

Example 4.9. List Domain Details Response: JSON

```
{
  "domains" : {
    "domain" :  {
      "id" : 2610368,
      "name" : "demodomainwithsubs.com",
      "ttl" : 5600,
```

```
    "emailAddress" : "demo@enduser.com",
    "nameservers" : {
      "nameserver" : [ {
        "name" : "ns.rackspace.com"
      }, {
        "name" : "ns1.rackspace.com"
      } ]
    },
    "records" : {
      "record" : [ {
        "id" : 2610368,
        "type" : "SOA",
        "name" : "demodomainwithsubs.com",
        "data" : "ns1.rackspace.com demo@enduser.com 3600 300
3600 3600 1298349924",
        "ttl" : 5600
      }, {
        "id" : 6012767,
        "type" : "NS",
        "name" : "demodomainwithsubs.com",
        "data" : "dns1.stabletransit.com",
        "ttl" : 5600
      }, {
        "id" : 6012768,
        "type" : "NS",
        "name" : "demodomainwithsubs.com",
        "data" : "dns2.stabletransit.com",
        "ttl" : 5600
      }, {
        "id" : 6012769,
        "type" : "NS",
        "name" : "demoDomainWithSubs.com",
        "data" : "ns1.rackspace.com",
        "ttl" : 86400
      }, {
        "id" : 6012770,
        "type" : "NS",
        "name" : "demoDomainWithSubs.com",
        "data" : "ns2.rackspace.com",
        "ttl" : 86400
      }, {
        "id" : 3012556,
        "type" : "MX",
        "name" : "demoDomainWithSubs.com",
        "data" : "mail.demoDomainWithSubs.com",
        "ttl" : 5000,
        "priority" : 2
      }, {
        "id" : 6500237,
        "type" : "A",
        "name" : "demoDomainWithSubs.com",
```

```
          "data" : "172.11.122.24",
          "ttl" : 86400
        } ]
      },
      "subdomains" : {
        "domain" : [ {
          "id" : 2610369,
          "name" : "sub.demodomainwithsubs.com",
          "ttl" : 5600,
          "emailAddress" : "hostmaster@secondlevel.com",
          "nameservers" : {
            "nameserver" : [ {
              "name" : "ns.rackspace.com"
            }, {
              "name" : "ns1.rackspace.com"
            } ]
          },
          "records" : {
            "record" : [ {
              "id" : 2610369,
              "type" : "SOA",
              "name" : "sub.demodomainwithsubs.com",
              "data" : "ns1.rackspace.com
hostmaster@secondlevel.com 3600 300  3600 3600 1298349928",
              "ttl" : 5600
            }, {
              "id" : 6012771,
              "type" : "NS",
              "name" : "sub.demodomainwithsubs.com",
              "data" : "dns1.stabletransit.com",
              "ttl" : 5600
            }, {
              "id" : 6012772,
              "type" : "NS",
              "name" : "sub.demodomainwithsubs.com",
              "data" : "dns2.stabletransit.com",
              "ttl" : 5600
            }, {
              "id" : 6012773,
              "type" : "NS",
              "name" : "sub.demoDomainWithSubs.com",
              "data" : "ns1.rackspace.com",
              "ttl" : 86400
            }, {
              "id" : 6012774,
              "type" : "NS",
              "name" : "sub.demoDomainWithSubs.com",
              "data" : "ns2.rackspace.com",
              "ttl" : 86400
            }, {
              "id" : 6500238,
```

```
              "type" : "A",
              "name" : "sub.demoDomainWithSubs.com",
              "data" : "172.11.122.23",
              "ttl" : 86400
            } ]
          }
        } ]
      }
    }
  }
}
```

4.1.3. Create Domain(s)

Verb	URI	Description
POST	/domains	Create a new domain with the configuration defined by the request.

Normal Response Code(s): 200, 202,

Error Response Code(s): dnsFault (400, 500), domainExistsFault (409), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation provisions one or, if subdomains are specified, multiple new DNS domains under the account specified, based on the configuration defined in the request object. If the corresponding request cannot be fulfilled due to insufficient or invalid data, an HTTP 400 (Bad Request) error response will be returned with information regarding the nature of the failure in the body of the response. Failures in the validation process are non-recoverable and require the caller to correct the cause of the failure and **POST** the request again.

Note

This process allows multiple records to be created along with the domain. This is an all-or-nothing success case: if there is a failure in creation of even a single record, the entire process will fail.

Example 4.10. Create Domain(s) Request: XML

```
http://localhost/cxf/dns/434705/domains/
<domains xmlns="http://docs.rackspacecloud.com/dns/api/v1.0">
    <domain  name="secondlevel434705.com" ttl="5600"
 emailAddress="hostmaster@secondlevel434705.com" >
        <records>
            <record type="A" data="172.11.122.24"
 name="secondlevel434705.com"/>
            <record type="MX" data="mail.secondlevel434705.com"
 priority="2" ttl="5000" name="secondlevel434705.com"/>
        </records>
    </domain>
</domains>
```

Example 4.11. Create Domain(s) Response: XML

```
{
   "domains":{
      "domain":{
         "name":"secondlevel434705.com",
         "accountId":"434705",
         "id":"2610502",
         "records":{
            "record":[{
               "ttl":"86400",
               "data":"172.11.122.24",
               "name":"secondlevel434705.com",
               "type":"A",
               "id":"6500343"
            },{
               "priority":"2",
               "ttl":"5000",
               "data":"mail.secondlevel434705.com",
               "name":"secondlevel434705.com",
               "type":"MX",
               "id":"3012593"
            }]
         }
      }
   }
}
```

Example 4.12. Create Domain(s) Request: JSON

```
{
   "domains":{
      "domain":{
         "name":"secondlevel434705.com",
         "accountId":"434705",
         "id":"2610502",
         "records":{
            "record":[{
               "ttl":"86400",
               "data":"172.11.122.24",
               "name":"secondlevel434705.com",
               "type":"A",
               "id":"6500343"
            },{
               "priority":"2",
               "ttl":"5000",
               "data":"mail.secondlevel434705.com",
               "name":"secondlevel434705.com",
               "type":"MX",
```

```
                "id":"3012593"
            }]
        }
    }
}
```

4.1.4. Modify Domain(s)

Verb	URI	Description
PUT	/domains/*domainId*	Modify the configuration of a domain.
PUT	/domains	Modify the configurations of multiple domains.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation modifies DNS domain(s) attributes only. Records cannot be added, modified, or removed. All attributes of a domain can be modified EXCEPT its `id`, `name`, and `serial number`.

Note

A domain can be modified only if the domain exists.

If a request cannot be fulfilled due to insufficient or invalid data, an HTTP 400 (Bad Request) error response will be returned with information regarding the nature of the failure in the body of the response. Failures in the validation process are non-recoverable and require the caller to correct the cause of the failure and POST the request again.

Note

A domain's `id` is never modifiable.

Example 4.13. Modify Domain Request: XML

```
http://localhost/cxf/dns/434705/domains/34454
<domain xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
 ttl="9500" emailAddress="hostmaster@newzone.com" />
```

Example 4.14. Modify Domain Request: XML

```
http://localhost/cxf/dns/434705/domains/
<domains xmlns="http://docs.rackspacecloud.com/dns/api/v1.0">
    <domain id="2610338" ttl="9200"
 emailAddress="hostmaster@newzone.com">
    </domain>
    <domain id="2610428" ttl="8800"
 emailAddress="changed@cloneddomain.com">
    </domain>
</domains>
```

4.1.5. Remove Domain(s)

Verb	URI	Description
DELETE	/domains/*domainId*	Remove a domain from an account.
DELETE	/domains/?id=*domainId1*&id=*domainId2*	Remove multiple domains from an account.
DELETE	/domains/*domainId*?deleteSubdomains=true	Remove a domain and its subdomains from an account.
DELETE	/domains/?id=*domainId1*&id=*domainId2* &deleteSubdomains=true	Remove multiple domains and their subdomains from an account.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation removes one or more specified domains from the account; when a domain is deleted, its immediate resource records are also deleted from the account. By default, if a deleted domain had subdomains, its subdomains become root domain(s) and are not deleted; this can be overridden by the optional `deleteSubdomains` parameter. When a domain is deleted, any and all configuration data is immediately purged and is not recoverable via the API. A backup of the configuration data may exist, but retrieving it would require manual intervention. In a request to remove multiple domains, a failure on a single part of the request will cause the entire request to fail. Utilizing the optional `deleteSubdomains` parameter on domains without subdomains does not cause a failure.

Example 4.15. Delete Domain Request: XML

```
http://localhost/cxf/dns/v1.0/434705/domains/344344
```

Example 4.16. Delete Domains Request: XML

```
http://localhost/cxf/dns/v1.0/434705/domains?id=344344&id=232323
```

Example 4.17. Delete Domain and Subdomains Request: XML

```
http://localhost/cxf/dns/434705/domains/344344?
deleteSubdomains=true
```

Example 4.18. Delete Domains and Subdomains Request: XML

```
http://localhost/cxf/dns/434705/domains?
id=344344&id=232323&deleteSubdomains=true
```

4.2. Records

4.2.1. List Records

Verb	URI	Description
GET	/domains/*domainId*/records	List all records configured for the domain. SOA cannot be modified.
GET	/domains/*domainId*/records/*recordId*	List details for a specific record.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

Example 4.19. List Records Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/records
```

Example 4.20. List Records Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<records xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
    xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <record ttl="5400" data="dns1.stabletransit.com"
 name="bbunittestacct2static3.com"
        type="NS" id="6012691" />
    <record ttl="5400" data="dns2.stabletransit.com"
 name="bbunittestacct2static3.com"
        type="NS" id="6012692" />
    <record ttl="9500"
        data="ns1.rackspace.com hostmaster@newzone.com 1299008840"
 name="bbunittestacct2static3.com"
        type="SOA" id="2610338" />
</records>
```

Example 4.21. List Records Response: JSON

```
{
  "records" : {
    "record" : [ {
      "id" : 6012691,
      "type" : "NS",
      "name" : "bbunittestacct2static3.com",
      "data" : "dns1.stabletransit.com",
      "ttl" : 5400
    }, {
      "id" : 6012692,
      "type" : "NS",
      "name" : "bbunittestacct2static3.com",
      "data" : "dns2.stabletransit.com",
      "ttl" : 5400
    }, {
      "id" : 2610338,
      "type" : "SOA",
      "name" : "bbunittestacct2static3.com",
      "data" : "ns1.rackspace.com hostmaster@newzone.com
 1299008840",
      "ttl" : 9500
    } ]
  }
}
```

Example 4.22. List Record Details Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/
records/6012691
```

Example 4.23. List Record Details Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<records xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
    xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <record ttl="5400" data="dns1.stabletransit.com"
 name="bbunittestacct2static3.com"
        type="NS" id="6012691" />
</records>
```

Example 4.24. List Record Details Response: JSON

```
{
  "records" : {
    "record" : [ {
      "id" : 6012691,
      "type" : "NS",
      "name" : "bbunittestacct2static3.com",
      "data" : "dns1.stabletransit.com",
      "ttl" : 5400
    } ]
  }
}
```

4.2.2. Add Records

Verb	URI	Description
POST	/domains/*domainId*/ records/*recordId*	Add new record(s) to the domain.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

Example 4.25. Add Records Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/records/
<records xmlns="http://docs.rackspacecloud.com/dns/api/v1.0">
    <record type="CNAME" data="dev.bbunittestacct2static3.com"
 name="sub.bbunittestacct2static3.com"/>
    <record type="A" data="172.23.2.212"
 name="bbunittestacct2static3.com"/>
</records>
```

Example 4.26. Add Records Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<records xmlns="http://docs.rackspacecloud.com/dns/api/v1.0"
    xmlns:ns2="http://docs.rackspacecloud.com/dns/api/management/
v1.0">
    <record ttl="86400" data="dev.bbunittestacct2static3.com"
 name="sub.bbunittestacct2static3.com"
        type="CNAME" id="9573515" />
    <record ttl="86400" data="172.23.2.212"
 name="bbunittestacct2static3.com"
        type="A" id="6500361" />
</records>
```

Example 4.27. Add Records Response: JSON

```
{
  "records" : {
    "record" : [ {
      "id" : 9573517,
      "type" : "CNAME",
      "name" : "sub.bbunittestacct2static3.com",
      "data" : "dev.bbunittestacct2static3.com",
      "ttl" : 86400
    }, {
      "id" : 6500365,
      "type" : "A",
      "name" : "bbunittestacct2static3.com",
      "data" : "172.23.2.212",
      "ttl" : 86400
    } ]
  }
}
```

4.2.3. Modify Records

Verb	URI	Description
PUT	/domains/*domainId*/ records/*recordId*	Modify the configuration of a record in the domain.
PUT	/domains/records/*recordId*	Modify the configuration of records in the domain.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

Example 4.28. Modify Record Configuration Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/
records/6500365/
<record data="172.23.2.214" name="bbunittestacct2static3.com"
 ttl="4544" type="A">
```

Example 4.29. Modify Records Configuration Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/records/
<record data="172.23.2.214" name="bbunittestacct2static3.com"
 ttl="4544" type="A">
    <records xmlns="http://docs.rackspacecloud.com/dns/api/v1.0">
        <record xmlns="http://docs.rackspacecloud.com/dns/api/
v1.0"
            data="172.23.2.214" name="bbunittestacct2static3.com"
 ttl="4544" type="A">
        </record>
    </records>
```

4.2.4. Remove Records

Verb	URI	Description
DELETE	/domains?id=*domainId*/records/*recordId*	Remove a record from the domain.
DELETE	/domains/*domainId*/records?id=*recordId1*&id=*recordId2*	Remove multiple records from the domain.

Normal Response Code(s): 200, 202

Error Response Code(s): dnsFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

Example 4.30. Remove Record Request: XML

```
http://localhost/cxf/dns/v1.0/440369/domains/2610338/
records/9573517
```

Example 4.31. Remove Records Request: XML

```
http://localhost/cxf/dns/440369/domains/2610338/records?
id=9573516&id=6500362
```

4.3. Record Types

4.3.1. List DNS Record Types

Verb	URI	Description
GET	/domains/resource_record_types	List all supported record types.

All domains must support certain record types. This list returns the currently supported record types and optional parameters. All records support the following four parameters: `name` (hostname), `data`, `type`, `TTL`. Optional parameters, when appropriate, are `priority` and `weight`. Unless otherwise noted and even if not specified, `name` is the fully-qualified domain name to which the record applies. When appropriate, example data may be included to clarify format.

Example 4.32. List Record Types Response: XML

```
<resource_record_types xmlns="http://docs.rackspacecloud.com/dns/
api/v1.0">
    <resource_record_type type="A" description="data: ip4 address,
 name:domain name, TTL: time to live" />
    <resource_record_type type ="CNAME" description="data:alias,
 name:domain name, TTL: time to live" />
    <resource_record_type type ="SOA" description="critical record
 only created and modified via add/modify domain calls" />
    <resource_record_type type ="NS" description="data: name
 server , name:domain name, TTL: time to live" />
    <resource_record_type type ="TXT" description="data:free form
 text , name:domain name, TTL: time to live" />
    <resource_record_type type ="PTR" description="a description
 of the record type's params" />  /* Currently unavailable */
    <resource_record_type type ="AAAA" description="data: ip v6
 address, name:domain name, TTL: time to live" />
    <resource_record_type type ="MX" description="data:mx server,
 name:domain name, TTL: time to live, priority: priority" />
    <resource_record_type type ="SRV" description="data:
 servers for specified service server:port ex:
 sipserver.example.com:5060, name: _service._proto.domainname
 ex: _sip._tcp.example.com, TTL: time to live, weight: weight,
 priority: priority"/>
</resource_record_types>
```